Information and Software Technology xxx (2011) xxx–xxx

Contents lists available at ScienceDirect

Information and Software Technology

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

Development and evaluation of a lightweight root cause analysis method (ARCA method) – Field studies at four software companies

Timo O.A. Lehtinen *, Mika V. Mäntylä, Jari Vanhanen

Department of Computer Science and Engineering, School of Science, Aalto University, P.O. BOX 19210, FI-00076 Aalto, Finland

ARTICLE INFO

Article history:
Received 17 December 2010
Received in revised form 3 May 2011
Accepted 15 May 2011
Available online xxxx

Keywords:
Root cause analysis
Problem prevention
Software process improvement
Industrial field study
Design science research
Cause-effect diagram

ABSTRACT

Context: The key for effective problem prevention is detecting the causes of a problem that has occurred. Root cause analysis (RCA) is a structured investigation of the problem to identify which underlying causes need to be fixed. The RCA method consists of three steps: target problem detection, root cause detection, and corrective action innovation. Its results can help with process improvement.

Objective: This paper presents a lightweight RCA method, named the ARCA method, and its empirical evaluation. In the ARCA method, the target problem detection is based on a focus group meeting. This is in contrast to prior RCA methods, where the target problem detection is based on problem sampling, requiring heavy startup investments.

Method: The ARCA method was created with the framework of design science. We evaluated it through field studies at four medium-sized software companies using interviews and query forms to collect feedback from the case attendees. A total of five key representatives of the companies were interviewed, and 30 case participants answered the query forms. The output of the ARCA method was also evaluated by the case attendees, i.e., a total 757 target problem causes and 124 related corrective actions.

Results: The case attendees considered the ARCA method useful and easy to use, which indicates that it is beneficial for process improvement and problem prevention. In each case, 24–77 target problem root causes were processed and 13–40 corrective actions were developed. The effort of applying the method was 89 man-hours, on average.

Conclusion: The ARCA method required an acceptable level of effort and resulted in numerous high-quality corrective actions. In contrast to the current company practices, the method is an efficient method to detect new process improvement opportunities and develop new process improvement ideas. Additionally, it is easy to use.

1. Introduction

Analyzing problem causes is considered in various software process improvement models, e.g., CMMI, ISO/IEC 12207, and Six Sigma [1]. The key for effective problem prevention is to know why the problem occurs [2]. We believe this is mainly because the reoccurrence of the problem can be prevented only through the elimination of its causes. Root cause analysis (RCA) is a structured investigation of a problem to identify which underlying causes need to be fixed [3]. It can help with process improvement and problem prevention in various contexts [1,4–12] and across all software organizations, including product development, hardware design, product engineering, and manufacturing [6].

Most of the reported industrial cases in software engineering root cause analysis [5,8,13–15] have aimed to lower defect rates by preventing the causes of the most typical types of the defects. The results are promising: a 50% decrease in defect rates [15], a 53% savings in costs and a 24% increase in productivity [13] has been indicated. However, the high number of particular types of software defects is not the only target problem that should be analyzed; e.g., negative project experiences [4], delayed product releases, and challenging product installations are all industrially relevant and severe problems but have only been exiguously explored using RCA.

There are many RCA methods [1–6,8,11,13,15–21], but no studies have included extensive analyses of the participants' feedback on the RCA method, and only a few studies have discussed the effort required to apply the method. Grady [8] indicates that 7 h of team work is the minimum cost of executing a non-recurring RCA method, whereas Mayes [6] indicates that the costs of the RCA method in large organizations consist of 8–10 action team members using 10% of their time for action team duties and 4–7 developers participating in kickoff and causal analysis meetings, each lasting 2 h. Card [15] indicates that the costs of the RCA

* Corresponding author. Tel.: +358 40 775 2781; fax: +358 9 470 24958.
E-mail addresses: timo.o.lehtinen@aalto.fi (T.O.A. Lehtinen), mika.mantyla@aalto.fi (M.V. Mäntylä), jari.vanhanen@aalto.fi (J. Vanhanen).

0950-5849/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.infsof.2011.05.005

method range from 0.5% to 1.5% of the software budget, which additionally requires a startup investment to fund the supportive infrastructure, i.e. defect classification scheme definitions, procedure setup, establishment of data collection mechanisms, and personnel training. Unfortunately, the prior RCA studies are too general to assess and compare the required startup and execution efforts in concrete man-hours. Additionally, most of the industrial RCA studies [5,6,8,13–15] were conducted at large software companies operating with mature development processes and products. The optimal RCA method for small- to medium-sized companies operating closer to a style of agile software development is likely to be different from the RCA methods presented in the prior studies.

This paper presents a lightweight RCA method and its empirical evaluation. For our research purposes, we developed an RCA method, referred to as the ARCA method, and evaluated it. This was done using a framework similar to that of design science [22,23], presented in Fig. 1. The environment of the research was the context of applying root cause analysis methods in software engineering. The business need was to develop a lightweight RCA method feasible for problem prevention at medium-sized software companies. The knowledge base of the method design was established by a literature review of root cause analysis (see Sections 2 and 3). The assessment of the design (see Sections 5 and 6) was performed through industrial field studies (see Section 4) by piloting the ARCA method in target problems of four medium-sized software product companies.

The research goal is as follows: To develop a lightweight RCA method for medium-sized software companies and evaluate it in industrial cases.

The ARCA method consists of four steps, i.e., target problem detection, root cause detection, corrective action innovation, and documentation of the results. Unlike the prior RCA methods applied in the software industry [5,6,8,13–15], the ARCA method does not require problem reports, e.g., software defect reports, in the target problem detection step. Instead, our method utilizes a focus group meeting to detect the target problem. This difference makes the ARCA method lightweight. It does not require heavy startup investment and, simultaneously, it is highly adaptable for various target problems.

We collected feedback from the case attendees to evaluate the easiness and usefulness of the ARCA method. Additionally, we measured the required effort and the output of the method, i.e., 757 target problem causes and their 124 related corrective actions. Even though implementing and monitoring the corrective actions is an important part of problem prevention programs [16], we excluded it from this research. It would have been practically impossible to separate the effects of the ARCA method from the company-specific context factors.

Evaluation of the method was conducted by answering the following research questions:

RQ1: *Is the ARCA method efficient?* "Efficiency" refers to the interrelationship between the advantages of the method output and the required effort. The output of the ARCA method is a set of corrective actions for the related root causes. "Quality of corrective actions" refers to their feasibility for and impact on the target problem.

RQ2: *Is the ARCA method easy to use?* "Ease of use" refers to the ease of conducting the steps of the ARCA method: target problem detection, root cause detection, corrective action innovation, and documentation of the results.

The rest of the paper is structured as follows. Section 2 discusses the theoretical background of root cause analysis. Section 3 introduces the ARCA method and its development. Section 4 presents the field study methodology used in the empirical part of this study. Section 5 shows the results of the field studies, and Section 6 answers the research questions and discusses the most interesting findings and threats to their validity. Section 7 states the conclusions and proposes future work on the topic.

2. Theoretical background

In this section, we introduce the framework of RCA methods. We first present definitions of root cause analysis in Section 2.1 and characterize what we mean by the word *target problem* in Section 2.2. Thereafter, in Section 2.3, we summarize the common steps of RCA methods and their related work practices.

2.1. Definitions of RCA

Usually, the idea behind RCA is to decrease the likelihood of a problem's reoccurrence [2,13,15,18], but, depending on the utilization context, RCA targets vary. For example, RCA is used to detect the causes of negative and positive project experiences [4] and to distill textual raw data, which is useful for requirement collection and knowledge elicitation [20].

There is no unique and commonly accepted definition for RCA [3,16] or for a root cause. Several authors introduce RCA as a cause

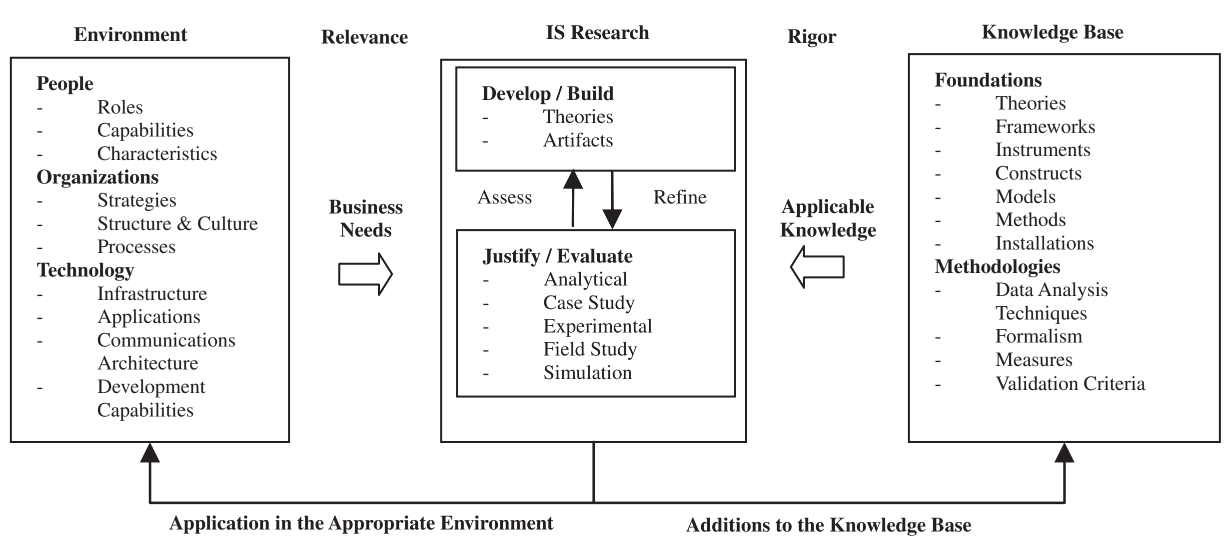

Fig. 1. Framework of design science in information systems research [22].

Please cite this article in press as: T.O.A. Lehtinen et al., Development and evaluation of a lightweight root cause analysis method (ARCA method) – Field studies at four software companies, Inform. Softw. Technol. (2011), doi:10.1016/j.infsof.2011.05.005

detection method only [2–4,13,17], whereas some authors present RCA as a problem prevention method that includes causal analysis and the development of corrective actions [11,15,16,18].

Some authors define a root cause as the deepest cause at the end of the causal structure [16,17], whereas others define it as any underlying cause of a target problem [2]. However, most of the authors recognize a root cause as a cause that management has the power to fix [2,16,17,21]. Logically, the target problem may have numerous root causes.

In our terminology, RCA is a process of detecting a target problem, collecting and organizing its causes, and recognizing its root causes. For our purposes, the RCA method for problem prevention means a method that includes RCA and the development of corrective actions. We define a root cause as any underlying cause of the target problem that the management has the ability to fix.

2.2. Target problem characterization

The target problem of RCA is a state of difficulty resulting in unwanted situations or events [16]. Additionally, it should be systematic and create severe consequences [1,5,8,11,13,15]. We believe that there is a wide variety of systematic target problems in software engineering, e.g., a defect, a high number of software defects, an overrun project budget, a late product release, a challenging product installation, lack of software testing, etc. Logically, the events and problems of a software company are interconnected with a cause and effect relationship. We describe this relationship as the interconnectivity of a problem.

It is claimed that every problem has a solution space that can be characterized in its complexity and cross-functionality [24]. The solution space of a problem corresponds to the number of solutions that the problem may permit [24]. The complexity of a problem corresponds to the effort required in its solution space to solve it, and cross-functionality represents the diversity of expertise required by the problem to attain solutions within its solution space [24]. We believe that the interconnectivity of the target problem correlates with the complexity and cross-functionality of its solution space.

An example of a target problem with a complex and cross-functional solution space is a late product release. The target problem may be caused by various difficult causes, e.g., overly optimistic schedule estimations, a large number of software defects, misunderstood requirements, or other unknown factors. A software defect alone may not be considered a severe situation. However, it may be caused by some systematic working methods in the development work; e.g., the features require modifications to a database, but developers omit them in 30% of the cases due to their busy schedules. As time goes on, the working methods might result in a large number of software defects in the project, causing delays to the already busy development schedule and, in the worst case, causing a delay in the product release. The busy development schedule, in turn, may be caused by various causes, not only because of the high number of the defects but also because of the overly optimistic project plans driven by misunderstood functional requirements established by the company sales personnel. We see that solving the target problem requires managing all of its interconnections. Unfortunately, some of the interconnections may not be controlled or prevented.

2.3. Common steps of RCA methods and related work practices

We found three steps that are common to the RCA methods introduced in the literature: (1) target problem detection [1–5,8,11,13,15–21], which defines the target problem of the RCA method, (2) root cause detection [1–5,8,11,13,15–21], which detects and organizes the causes of the target problem, and (3) corrective action innovation [1–3,5,8,11,13,15–19], which develops corrective actions for the most important root causes. Alternative work practices have been presented for each of the above steps. These are presented in Sections 2.3.1–2.3.3.

2.3.1. Target problem detection

A target problem for RCA is detected through problem sampling [1,3,5,8,13,15,16,19], interviewing [2,3,25], brainstorming [3,16], and flowcharting [3,16,17]. Usually, there is a meeting where the target problem is finally decided upon [15,19].

The usual case of software engineering root cause analysis exploits the Pareto principles to software defects to detect the target problem [1,5,8,11,13,15,18]. The idea is to sample and classify software defects and, thereafter, to select the class containing the highest number of defects as the target problem.

Another approach to detect the target problem is using qualitative methods, such as interviewing [3,25] software development managers to name the main problems of the development work. The negative side of this approach is that the target problem might not be as focused as it might be if exploiting Pareto principles to software defects. The positive side is that this is a quick and easy way to select an important and severe problem for analysis. Thus, the large workload of defect classification and deeper analyses can be avoided. For some organizations, there is a great motivation to use this approach, as there is little possibility of separate resources for the RCA investigation, while there are more possibilities in a large company [6,8]. The mad schedule rush of software companies forces them to progress in new projects rather than focusing on analyzing the defects of yesterday's projects [26]. However, in a large development effort, this approach could cause "too many" target problem causes to be detected, so the magnitude of work to analyze all the relevant causes would stay high [5,8,13].

2.3.2. Root cause detection

In root cause detection, there are different ways to collect and organize the target problem causes [25]. The causes are usually collected from various stakeholders [3,15,19,25] using interviewing [17], questionnaire [16,27], brainstorming, and brainwriting methods [3,16,27]. The questionnaires and interviews are more anonymous approaches, in contrast to the brainstorming and brainwriting approaches, which are performed publicly.

The target problem causes are usually organized into a cause-effect diagram based on their cause and effect relationships using a fishbone diagram [4,11,16,19,28], a fault tree diagram [16], a causal map [4], a matrix diagram [16], a scatter chart [16], a logic tree [3], or a causal factor chart [2]. It has been shown that lists, worksheets, and charts may also be used to organize the causes [17]. The root causes are finally detected by analyzing the collected target problem causes by focusing on the causes that will be prevented [2,15].

2.3.3. Corrective action innovation

Corrective actions are usually developed in a meeting [5,8,13,15,16], where brainstorming and brainwriting are the recommended work practices [16]. Brainstorming has three major obstacles that brainwriting can tackle: (1) people cannot speak simultaneously, (2) there is a fear of negative evaluation from other group members, and (3) individual contributions are not identifiable [4,16,24]. Additionally, it has been claimed that brainwriting is a feasible practice to address complex problems, whereas in cross-functional problems (see Section 2.2), brainstorming attains better solutions [24]. These practices can also be mixed with problem elimination techniques, such as Systematic Inventive Thinking, the Theory of Inventive Problem Prevention, or the Six Thinking Hats [16]. However, these techniques are rather complex, and more creative approaches should be used [16].

Please cite this article in press as: T.O.A. Lehtinen et al., Development and evaluation of a lightweight root cause analysis method (ARCA method) – Field studies at four software companies, Inform. Softw. Technol. (2011), doi:10.1016/j.infsof.2011.05.005

3. Description of the ARCA method

In this section, we present the ARCA method. First, in Section 3.1, we introduce how the method was developed. Thereafter, in Section 3.2, we present the work phases and practices of the ARCA method and compare these to the most notable prior RCA methods by following the common steps of the RCA methods introduced in Section 2.3.

3.1. Development of the ARCA method

We started the design of the ARCA method by setting down its requirements. We believe that a beneficial, lightweight RCA method would help software companies to develop high-quality corrective actions with low effort. We think that this goal can be satisfied by fulfilling the following requirements:

1. Helps to develop corrective actions that are feasible and effective for solving the target problem.
2. Requires low effort.
3. Is easy to use.
4. Is adaptable for different kinds of target problems.

Thereafter, we performed a literature review. The literature included RCA methods used in the software industry and also in other contexts. The literature was collected using predefined search words (“RCA,” “root cause analysis,” “DCA,” “Defect Causal Analysis,” “defect analysis,” “defect prevention,” and “problem prevention”) in Google and Scopus. The review was driven by the following questions:

1. Are there steps common to all RCA methods?
2. What are the recommended work practices in the different steps of RCA?

We designed an initial version of the ARCA method based on its requirements and the literature review. During the method design, we performed an analytical argumentation on various alternatives introduced in prior works. The initial ARCA method was piloted with a student software project. This was very important because it made it possible to refine the method before the industrial field studies were conducted. For example, we realized that a monitor and a software tool should be used to visualize and register the problem causes because of the high number of them. Using Post-it notes was unfeasible for this purpose.

3.2. The ARCA method

In this section, we first introduce the most notable prior RCA methods. Thereafter, in Sections 3.2.1–3.2.4, we describe in detail the work phases and practices of the ARCA method and argue the design by comparing it to these prior RCA methods. We discuss these methods because they were presented in enough detail in the related publications and, like the ARCA method, they follow the common steps of the RCA methods, as summarized in Table 1.

Rooney and Vanden Heuvel [2] present an RCA method consisting of four work phases: data collection, causal factor charting, root cause identification, and recommendation generation. The method starts with the data collection, where the team gathers a target problem’s related data. In the causal factor charting, the team organizes and analyzes the results of the data collection. Causal factor charting provides a way to structure the data based on its cause and effect relationships to a sequence diagram, which helps investigators to recognize *causal factors* that are seen as the most likely potential causes of the target problem. Thereafter, in the root cause identification, the investigators analyze the causal factors using a decision diagram, which is an up-front collection of potential problem causes helping to answer questions about why a particular cause exists. Finally, in the recommendation generation, the

Table 1
Summary of the ARCA and prior RCA methods and their work phases.

RCA method	Target problem detection step		Root cause detection step		Corrective action innovation step	
	Work phase	Work practices	Work phase	Work practices	Work phase	Work practices
Rooney and Vanden Heuvel [2]	Data collection	Interviewing, inspections	Causal factor charting	Sequence diagram	Recommendation generation	–
			Root cause identification	Decision diagram		
Ammerman [17]	Problem definition and data collection	–	Event and causal factor charting	Sequence diagrams	Corrective action development	Interviewing
	Task analysis	Paper-and-pencil, walk-through	Root cause determination	Interviewing, event and causal factor charts, lists, and worksheets		
	Change analysis	Flow charts				
	Control barrier analysis	Flow charts				
Latino and Latino [3]	Opportunity analysis	Sequence diagrams, interviewing, Pareto analysis	Data analysis	Flow chart, logic tree, meetings	Recommendation development	Writing individually, meetings
Card [15]	Defect sampling	Sampling, meetings	Determining principal cause	A fishbone diagram, cause categories, meetings	Development of action proposals	Meetings
	Defect classification	Classification scheme, meetings				
	Identifying systematic errors	Pareto analysis, meetings				
ARCA method	Target problem detection	A focus group meeting	Preliminary cause collection	Anonymous email inquiry, a directed graph	Root cause selection	Email inquiry
			Causal analysis workshop	Brainwriting and brainstorming in a meeting, a directed graph	Corrective action workshop	Brainwriting combined with skeptical and optimistic perspectives in a meeting

Please cite this article in press as: T.O.A. Lehtinen et al., Development and evaluation of a lightweight root cause analysis method (ARCA method) – Field studies at four software companies, Inform. Softw. Technol. (2011), doi:10.1016/j.infsof.2011.05.005

investigators develop corrective actions for the most important causal factors.

Ammerman [17] introduces an RCA method (PIC) consisting of eight work phases: problem definition and data collection, task analysis, change analysis, control barrier analysis, event and causal factor charting, root cause determination, corrective action development, and reporting conclusions. The method starts with defining a target problem, which is followed by collecting the problem-related data. The task analysis helps the team to understand where the pitfalls are within the target problem that is under evaluation. The goal is to find out what was assumed to have happened, not exactly what happened. Instead, the change analysis helps to understand what actually happened and what was expected to happen. The activity that was successfully performed is compared to an activity that was unsuccessfully performed. The focus of the control barrier analysis is to discover where physical or administrative barriers are needed to prevent the target problem. In the event and causal factor charting, a flow chart that graphically displays an entire event resulting in the target problem is created. The work phase of the root cause determination aims to detect the root causes of the target problem. The team should detect the root causes in a systematic way and utilize visual tools such as lists, worksheets, and charts. The goal of the corrective action development is to identify, develop, and evaluate corrective actions required to prevent the target problem's recurrence or significantly reduce its likelihood. Finally, the team documents all the intermediate results and recommended corrective actions.

Latino and Latino [3] present an RCA method (PROACT) consisting of four work phases: opportunity analysis, data analysis, developing recommendations, and reporting conclusions. In opportunity analysis, failures are sampled and classified. Then, Pareto analysis is used to detect the most likely potential target problems for RCA. Thereafter, in the data analysis, cause and effect relationships are detected and structured using a logic tree, which is a combination of a logic diagram and a fault tree. The goal is to detect the root causes of the target problem by listing and structuring hypothetic causes and either proving or disproving them with hard data. In corrective action development, the team first decides on an acceptance criterion for recommendations. Thereafter, the team develops recommendations to address the target problem root causes. Finally, the team documents all the findings, including the failures, root causes, and recommendations.

Card [15] presents Defect Causal Analysis (DCA), an RCA method consisting of six work phases: defect sampling, defect classification, identifying systematic errors, principal cause determination, developing action proposals, and reporting conclusions. In defect sampling, software defects are sampled to explore those that occur most frequently and have the most negative impact on the quality of the software. Thereafter, in the defect classification, investigators identify clusters of software defects by classifying the sample. Then, they use Pareto analysis to identify systematic defects. In principal cause determination, the root causes of the systematic defects are detected. If the root cause is not obvious from the defect statement, it should be drawn out using a fishbone diagram. In the development of action proposals, the corrective actions are developed for the determined root causes to either detect systematic defects earlier or prevent them. Finally, all the results, including the root causes and corrective actions, are recorded.

3.2.1. Step 1: Target problem detection

This is the first step of the ARCA method. After this step, the target problem will have been defined. Rooney and Vanden Heuvel [2] and Latino and Latino [3] indicate that interviewing is a feasible practice in detecting the target problem. However, we emphasize a focus group meeting because it is an excellent approach to identify rapidly what is important to the people [29]. We also believe that it requires less effort than interviewing and is easy to conduct. Flow charting is also shown to be a useful method in problem detection [17]. However, the intangibility of software engineering problems makes it difficult to create flow charts to describe how they evolve.

We believe that problem sampling [3,15,18] is unfeasible for many target problems. It sounds like a great idea to analyze and eliminate the causes of the most usual type of problems to lower the likelihood of their reoccurrence. On the other hand, problem sampling requires effort and information that is not easily available in practice [30]. For example, our collaboration with industrial partners suggests that information such as the defect type or defect module is sporadically reported by the company's personnel [31], thus making the defect data too unreliable for RCA. Additionally, according to [32], it is labor-intensive and probably not worthwhile to link the defects to their causes in large development efforts, as it may not lead to ideas that can be used to improve the software engineering mechanisms. Moreover, the problem sampling can be done only for the problems that are reported [1,15,19,33], and, in many cases, defect databases do not contain problems such as requirements faults [33].

In the ARCA method, the first step starts with a focus group meeting where the target problem is defined and the causal analysis workshop participants, who are to collect the target problem causes and to evaluate root causes, are selected (4–10 participants). The RCA facilitator holds this meeting with company staff, e.g., the managers who are responsible for product quality. In the meeting, the following issues should be justified and documented: what is the target problem and why exactly is this problem important to prevent? When selecting the causal analysis participants, it is important to include target problem experts that represent different stakeholders around the target problem. These may include project managers, developers, testers, software quality assurance staff, product managers, and process improvement group members.

3.2.2. Step 2: Root cause detection

This is the second step of the ARCA method. After this step, the most important root causes will have been detected and evaluated. We see that both anonymous and public approaches are important in root cause detection. Anonymity encourages the participants to address causes that they believe are dangerous to say aloud, whereas publicity helps to address causes that many participants value highly. The other RCA methods do not emphasize this. Ammerman [17] emphasizes interviewing only, whereas Latino and Latino [3] and Card [15] emphasize meetings.

Unlike the prior RCA methods, we recommend using a directed graph [4] to structure the causes based on their cause-and-effect relationships (see Fig. 2). As the directed graph represents a network of causes, each cause needs to be placed only once in the cause-effect diagram. The cause-effect diagrams of the prior RCA methods result in the problem of duplicating the same cause multiple times if the cause simultaneously explains more than one cause. Card [15] recommends using a fishbone diagram, which he claims to be a simple technique. However, using the fishbone diagram does not solve the duplicating problem. The problem also occurs when a logic tree is used, which is recommended by Latino and Latino [3]. Rooney and Vanden Heuvel [2] recommends using a sequence diagram followed by a decision diagram. Unfortunately, the sequence diagram also includes the duplicating problem and the decision diagram includes the challenge of detecting the correct problem causes in advance, as the target problems vary. Additionally, we believe that using two diagrams is more difficult than using one. Ammerman [17] indicates that structuring the target problem causes should be done with visual tools such as lists, worksheets, and charts. However, it is likely that too many target

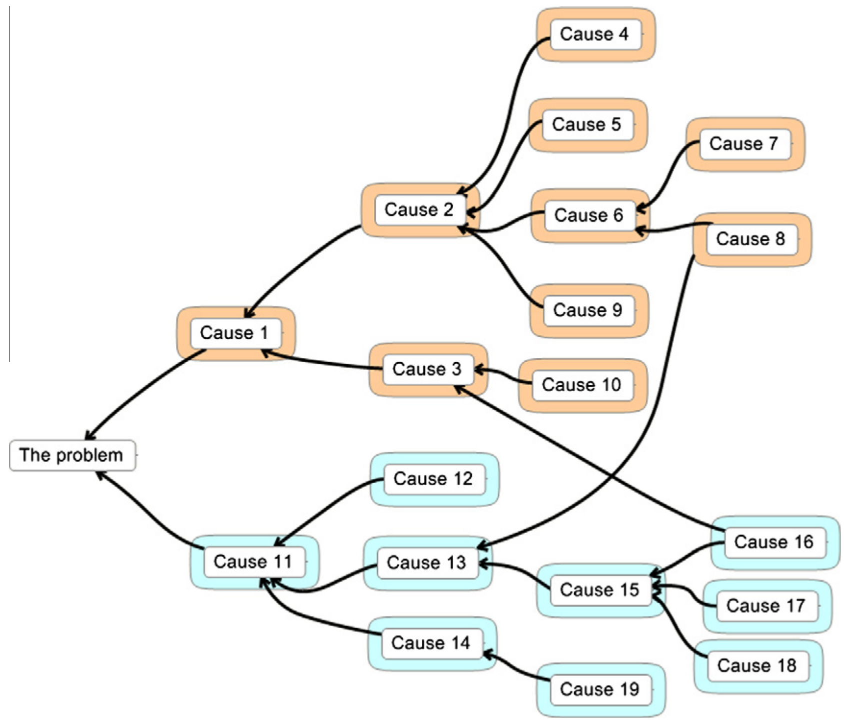

Fig. 2. The cause-effect diagram of the ARCA method.

problem causes will be detected to be visualized using these tools [5]. Additionally, the duplicating problem occurs with lists, worksheets, and charts.

In the ARCA method, the second step consists of two work phases: preliminary cause collection and a causal analysis workshop. In preliminary cause collection, the RCA facilitator sends out an email inquiry to the case participants and collects the target problem causes. The inquiry asks the participants to list at least five causes of the target problem. Since the listed causes probably complement one another, they are organized into a cause-effect diagram by the RCA facilitator, as presented in Fig. 2. Using a software tool is recommended here.

The second work phase is the causal analysis workshop, which is prepared by the RCA facilitator. A cause entity (see the colored causes in Fig. 2) includes a cause and its sub-causes, which together form an entity that is reasonable to process together. By analyzing the cause-effect diagram, the RCA facilitator selects the most important cause entities to be processed in the workshop. It is possible that the entities will overlap since the causes explain one another. Processing a cause entity containing about 10 causes can be done adequately in about 40 min. We recommend this as a suitable size for a cause entity.

The causal analysis workshop is a meeting where new target problem causes are collected and analyzed. The workshop has a recommended minimum duration of 40 min per cause entity. At the beginning of the workshop, the RCA facilitator presents the target problem, the preliminary causes, and the selected cause entities. Thereafter, new causes are collected for each selected cause entity. The cause entities are processed one at a time. Each cause can either deepen or widen a cause entity. Collecting the causes into a cause entity is done in three parts:

1. The participants write new causes down on paper for 5 min (the cause-effect diagram should be projected onto the wall).
2. Each participant presents the causes he or she has written and explains where they should be placed in the cause-effect diagram.
3. The participants briefly discuss the cause entity's causes, trying to brainstorm more causes and to recognize whether a cause has a relationship to other causes.

After all the selected cause entities have been processed, the related cause-effect diagram is analyzed as a whole. The RCA facilitator asks the participants to point out essential causes and to discuss them. The controllable causes, i.e., the root causes, are identified. The other causes are set aside and are not processed any further.

3.2.3. Step 3: Corrective action innovation

This is the third step of the ARCA method. After this step, the corrective actions for the most important root causes will have been developed. In the prior RCA methods, there is very little practical guidance on how to develop corrective actions. Keeping a meeting where the corrective actions are developed is presented by Latino and Latino [3] and Card [15], whereas Ammerman presents interviewing techniques to be used [17]. We believe that keeping the meeting helps to develop commitment to the corrective actions among the participants more than the interviewing techniques. In the corrective action innovation, we chiefly emphasize brainwriting because it provides an efficient way to use all of the participants simultaneously. However, as we believe that there are also advantages in brainstorming (see Section 2.3), we recommend it to refine the findings into the best corrective actions. Latino emphasizes brainstorming in the corrective action innovation but stresses also that it is important to write down the corrective actions [3]. Ammerman indicates that it is important to develop multiple corrective actions and to evaluate and select them to have alternatives [17]. We found that the commonality between the elimination techniques presented in the literature [16] is that a corrective action is analyzed from different perspectives, especially from optimistic and skeptical perspectives. Therefore, we adopted the idea of different perspectives to the ARCA method by creating a paper template for a corrective action that forces the participants to brainwrite the corrective actions from both perspectives (see Appendix E).

In the ARCA method, the third step consists of two work phases: the root cause selection and the corrective action workshop. The first work phase includes the selection of the root causes. To focus the available resources as efficiently as possible, the RCA facilitator has to carefully select the root causes for which corrective actions are to be developed. First, the finalized cause-effect diagram is sent to the participants of the causal analysis workshop. The participants are asked to propose root causes for which corrective actions should be developed and evaluate them using the following criteria: the level of impact on the target problem and the level of difficulty of developing corrective actions. Then, the RCA facilitator selects 4–6 root causes to be processed using his judgment and analysis of the root causes proposed by the participants. Finally, the RCA facilitator documents each of the selected root causes including its sub-causes into a cause-effect diagram, each for an individual paper.

The second work phase of the step is the corrective action workshop, which is a meeting wherein the corrective actions of the selected root causes are developed, evaluated, and analyzed. The workshop has a recommended duration of 2 h. First, the RCA facilitator selects 4–6 participants to join the workshop. They have to be an aggregate of experts who are as competent as possible at solving the selected root causes. In the workshop, each participant works, in turn, for 10–15 min with one root cause. They develop corrective actions by writing them down on paper (see Appendix E) and rotating them through the participants. The root causes are rotated until every participant has treated all the root causes. A participant can also supplement corrective actions developed by other participants by adjusting, expanding, and commenting on them. The corrective actions are evaluated to find the best corrective actions. The evaluation is conducted similarly to their development: the root causes, including their corrective actions, are rotated through the participants. Each participant evaluates corrective actions of a root cause by giving two attributes to each (scale of 1–5): impact on the target problem and feasibility. The last participant evaluating the corrective actions of a root cause calculates the sum of evaluations of each corrective action. Then, he presents the corrective action that has the highest value of the multiplication of the impact and feasibility. This is done for each processed root cause. The participants are asked to discuss the corrective action and to refine it. The presenter writes down the comments and improvement suggestions concerning the action he presented.

3.2.4. Step 4: Documentation of the results

During this final step of the ARCA method, the results are compiled into a final report, which includes at least the target problem definition, the cause-effect diagram, and all of the corrective actions, including their evaluations. This step is also mentioned by Card [15], Latino and Latino [3], and Ammerman [17]. The best corrective actions should be implemented to make the actual changes in the way of working. Because gaining currency for a corrective action can be challenging, the final report can be used to justify the changes required to prevent the target problem. Additionally, the final report can be a valuable source of cause information in future RCA cases.

4. Field study methodology

This section introduces the field study methodology [29] used in the empirical part of this study. Section 4.1 presents how the data collection and analysis was conducted in the field study settings. We introduce the data collection methods used, including their focus and how the collected data was analyzed. In Section 4.2, we introduce the industrial cases wherein the field studies were conducted.

4.1. Data collection and analysis

Triangulation of the data sources and the data collection methods increases the reliability of the results [34,35]. We used interviews [34], query forms [36], measurements, and observations [34] to collect empirical evidence from the industrial cases to evaluate the feasibility of the ARCA method. Table 2 summarizes the data collection methods and their focus in the analyses of this study. Sections 4.1.1–4.1.3 introduce these instruments in detail and discuss how they were used.

The data analysis was conducted in two phases. After each case, we analyzed the collected data to help understand the strengths and weaknesses of the ARCA method used in the current case. After all the cases were conducted, we evaluated the method as a whole by combining all empirical evidence from the industrial cases and comparing the results among the interviews, query forms, observations, and measurements.

4.1.1. Interviews

The key representatives were company people involved in steering the cases and had the power to make changes in their companies. Interviews were held with them before and after a case to analyze how they experienced the ARCA method in general. The researchers tested the interview questions with colleagues before the cases.

Interview 1 (see Appendix A) was a group interview with 2–4 company key representatives. Its goal was to give an overview of the case context (see Section 4.2). Interview 2 (see Appendix B)

Table 2
Targets of the data collection instruments.

Target	Presented in	Interview 1	Query form 1	Query form 2	Interview 2	Measurement	Observation
Case context	Case information	x					
Section 4.2	Current practices	x					
	Case participants		x	x			
	Case target problem	x	x				
RQ1	Number of detected causes (Table 5)					x	
	Cause correctness Tables 6 and 7)		x		x		
	Importance of the processed causes (Tables 6 and 7)			x	x		
	Number of processed causes (Table 5)					x	
	Number of corrective actions (Table 5)					x	
	Feasibility of the corrective actions (Fig. 3 and Tables 6 and 7)			x	x	x	
	Impact of the corrective actions (Fig. 3,Tables 6 and 7)			x	x	x	
	Effort used (Table 4)			x		x	
	Feasibility of the method (Tables 6 and 7)		x	x	x		x
RQ2	Easiness of the method (Tables 6 and 7)		x	x	x		x

was conducted with the key representatives responsible for steering the case. Its goal was to evaluate the practices and output of the ARCA method. Before interview 2 was conducted, the final report of the ARCA method (see Section 3.2.4) was first examined.

As Yin recommends [34], a similar protocol was used in each interview and the duration was no longer than 60 min. Each interview was recorded and transcribed by the first author. Thereafter, the answers were cleaned up and entered into an Excel sheet according to the following coding themes: case context, method usefulness, method easiness, and output quality. Finally, the particular theme was analyzed between the cases by comparing how the answers varied.

4.1.2. Query forms

The query forms were used after the causal analysis and corrective action workshops to analyze how the case participants experienced the ARCA method and its output. The query forms included closed and open-ended questions, as recommended by [36]. The researchers tested and reviewed the query forms with colleagues before using them. Additionally, they were tested with students who piloted the ARCA method before the industrial cases. We asked the names of the case participants in the query forms because we wanted to analyze how the answers of particular participants varied between the workshops. Unfortunately, it is possible that this slightly skewed the results, as the participants knew that the researchers might at least note their names. However, we stressed that the answers are confidential and emphasized the importance of giving feedback as straightforwardly as possible.

Query form 1 (see Appendix C) was designed to help in analyzing how the case participants experienced the case target problem and the work practices of the root cause detection step (see Section 3.2.2). Query form 2 (see Appendix D) was designed to help analyzing how the case participants experienced the importance of the processed root causes and the work practices of the corrective action innovation step (see Section 3.2.3). We also analyzed whether the output of the steps of the root cause detection and corrective action innovation was correct according to the case participants. Similarly, we analyzed the feasibility and impact of the corrective actions.

The data from the query forms was entered into an Excel spreadsheet to make it possible to analyze just one case or all the cases simultaneously. All the answers from each participant were divided into separate cells according to the coding themes presented in Section 4.1.1. For every quantitative question in the query forms, we calculated the averages and standard deviations of the answers for each case separately and for all the cases simultaneously.

4.1.3. Measurements

We measured the effort used and the output of the ARCA method. We kept an accurate record of how many man-hours were used in the different activities of the method and how many causes were detected and processed during the cases. We also kept an accurate record of how many corrective actions were developed and how the case participants evaluated the feasibility and impact of each corrective action with respect to the target problem.

The effort used was measured straightforwardly in most of the activities of the ARCA method, as we were able to video-record them. However, the effort used in two work phases relied on the reports of the case participants. Each case participant reported independently how much effort they used in the preliminary cause collection (see Section 3.2.2) and in proposing the root causes for which the corrective actions should be developed (see Section 3.2.3). The required effort for the ARCA method was entered into an Excel sheet to analyze how many man-hours were actually used in the different steps of the method and how many people contributed there.

The number of detected and processed target problem causes during the cases was measured straightforwardly. We divided the causes according to the steps of the ARCA method. Similarly, we were able to measure the number of corrective actions. During the data analysis, the number of target problem causes and the number of corrective actions were entered into an Excel sheet to compare the cases.

A paper template (see Appendix E) was used to develop the corrective actions and to evaluate their quality, as presented in Section 3.2.3. The paper template included an evaluation form that was used by the case participants to evaluate the feasibility and impact of each corrective action. During the data analysis, we compared the cases by analyzing the corrective actions based on these evaluations. The evaluation form was not anonymous, as the case participants were able to see what the others answered. Thus, it is possible that the evaluations were biased.

4.1.4. Observations

Two researchers participated in each case. One steered the case together with the key representatives, whereas one focused only on observing the actions during the video-recorded workshops. Both researchers wrote notes during the workshops. After each workshop, the researchers held a feedback session together. The observation data was used to confirm the results of the interviews and query forms on the feasibility and easiness of the work phases of the ARCA method.

4.2. Industrial cases

The field studies were conducted at four medium-sized software companies located in Finland. Based on interview 1, Sections 4.2.1–4.2.4 introduce these case companies and the related cases in detail. The target problem of the ARCA method was chosen by the key representatives of the case company, who also selected the case participants. To avoid the possibility that the cases could be highly different, the key representatives were requested to choose generally similar target problems, i.e., a complex software engineering problem that causes delays in software projects.

Table 3 summarizes the company cases with the data important for using the ARCA method. In the table, the qualitative data is based on interview 1, whereas the quantitative data is based on query form 1. Including the effort the company has expended trying to solve the target problem previously, the table summarizes how the key representatives characterized the target problems and how the case participants evaluated it. The impact evaluation of the target problem is a combination of the query form questions regarding the "impact of the target problem for the quality of the product," "adverse effect of the target problem to my daily work," "impact of the target problem for the end users of the product," "impact of the target problem for customer relationships," and "internal impact of the target problem for the company."

The similarities of the cases made them more comparable, whereas the dissimilarities consolidated the field study results in different case contexts. In each case, the target problem was experienced as highly complex and difficult to prevent. Similarly, in each case, the impact of the target problem was experienced as relatively high. Instead, the target problem itself and the effort the company had employed to try to prevent it varied between the cases and between the opinions of the case attendees. Additionally, the company size and the current company practices, including the available resources for software process improvement, varied. There were also differences in the roles of the case participants.

Please cite this article in press as: T.O.A. Lehtinen et al., Development and evaluation of a lightweight root cause analysis method (ARCA method) – Field studies at four software companies, Inform. Softw. Technol. (2011), doi:10.1016/j.infsof.2011.05.005

Table 3
Summary of the case contexts.

	Case 1	Case 2	Case 3	Case 4
Case company	Software company with 100 employees	Software company with 450 employees	Software company with 100 employees	Software company with 110 employees
Target problem	Fixing and verifying defects delays project schedules	Blocker type defects are detected in the product after release	New product installation and updating are challenging tasks	Issues' lead time is sometimes intolerably long
Roles of the case participants	Project managers, quality managers, developers, sales personnel, $N = 9$	Mostly developers, $N = 9$	Project managers, testers, developers, $N = 7$	Project managers, testers, developers, sales personnel, $N = 6$
Target problem characteristics	"*Extremely costly and complex*"	"*Not very costly, but very complex*"	"*High impact on customer relationships and complex*"	"*Extremely costly and complex*"
Difficulty of preventing the target problem[a]	Average = 5.3	Average = 5.6	Average = 5.4	Average = 5.5
	Standard deviation = 1.1	Standard deviation = 0.8	Standard deviation = 1.3	Standard deviation = 1.0
Earlier effort surrounding the target problem[a]	"*We have continuously tried to solve this*"	"*During recent months, we have reacted to this*"	"*We haven't managed this much*"	"*We have discussed how to improve communication*"
	Average = 3.0 Standard deviation = 1.0	Average = 4.3 Standard deviation = 1.3	Average = 3.4 Standard deviation = 0.5	Average = 3.0 Standard deviation = 0.6
Impact of the target problem[a]	Average = 5.8	Average = 5.0	Average = 5.6	Average = 5.9
	Standard deviation = 1.1	Standard deviation = 1.3	Standard deviation = 0.9	Standard deviation = 0.9

[a] Scale: 1 = very low; 2, 3, 4 = neutral; 5, 6, 7 = very high.

4.2.1. Case 1

The first case was conducted at Company 1, a medium-sized international software product company with approximately 100 employees. The average size of the project organization is about seven people. The main product is a large and complex software system, released twice a year, consisting of a major and a minor release.

The key representatives assumed that the company uses approximately 0.9% of its annual budget for software process improvement, which is managed by a quality assurance (QA) team consisting of three people. The QA team holds meetings in which different kinds of problems based on their criticality are selected and processed. The problems are initially detected by interviewing different stakeholders, such as project managers and product owners. The company's earlier experiences in RCA were fairly insignificant.

The target problem of the case was that the product releases are delayed due to a high number of software defects detected at the end of the development projects. The company has continuously tried to prevent the problem during recent years. The key representatives' common opinion was that the problem is extremely complex and costly for the company. They claimed that the main problem causes are that the size of technical blocks in the software is too large and that employees' attitudes are not fertile enough to develop high-quality software at once. Additionally, they assumed that increasing discipline among the developers and releasing the software in shorter cycles would help in eliminating the target problem.

4.2.2. Case 2

The second case was conducted at Company 2, a medium-sized international software product company with approximately 450 employees. The company releases new software versions regularly and its products can be characterized as complex and model-based software.

The key representatives assumed that approximately 1% of the annual budget of the company is used on software process improvement, which is divided into different levels of the company. While managers are asked to use 5–10 min daily to think about how the software process could be improved, the developers and requirements engineers are involved in process improvement meetings on a regular basis. Additionally, all detected defects are prioritized on a daily basis by a group of 15–20 people. The company used RCA earlier by applying a "five times why" practice in process improvement meetings.

The target problem of the case was that blocker-type defects are detected after the product releases, which increases the costs of redevelopment. The company has recently reacted to this problem by setting a clear goal to lower the number of defects detected by the customers. The key representatives characterized the target problem as very complex and including many different causes. The main causes for the target problem were believed to be the fact that new code is built on the old, low-quality code, too many different methods are used in the development work, and the lack of different hardware set-ups decreases the coverage of the software testing. They said that the problem could be best eliminated by refactoring the old code. They also believed that the problem is not very severe because the customers are currently highly satisfied.

4.2.3. Case 3

The third case was conducted at Company 3, a medium-sized international software product company with approximately 100 employees. The main product can be characterized as a highly configurable software service. The product is delivered for the customers through installation projects that occasionally include the development of new features. New software versions are released regularly.

The key representatives assumed that the company uses approximately 3–5% of its annual budget on software process improvement, which is managed by a quality manager, assisted by a quality management system. The project teams use weekly meetings in which positive and negative project experiences are discussed. The company's earlier experiences with RCA were fairly low.

The target problem of the case was that the installation projects are too challenging to be performed efficiently. It often follows that re-engineering has to be done because of unexpected defects caused by the complex software configurations and new development work during the projects. The company has not expended much effort to manage the target problem earlier. However, the key representatives stressed that the target problem has a significant impact on their customer relationships and that it is very complex to prevent. They said that the main cause of the target problem is that the employees have too many different ways in which to perform a product installation. Additionally, the number of different stakeholders is too high with respect to the quality of communication between them. They also indicated that the target problem could be minimized by creating checklists and simplifying the installation process.

4.2.4. Case 4

The fourth case was conducted at Company 4, a medium-sized international software product company with approximately 110 employees. The main product can be characterized as a highly complex software system. The product is delivered to customers through complex integration projects where the product is configured into the software systems of the customers.

The key representatives assumed that the company uses approximately 3–5% of its annual budget on software process improvement. The company's management team is responsible for writing process guidelines and for improving the software development process in general. Coding and testing teams are required to improve their daily work through regular process improvement meetings. The teams work together regularly. The company's earlier experiences in RCA were fairly insignificant.

The target problem of the case was that the lead time of an issue is occasionally intolerably long, resulting in delays in projects. The company has not expended much effort to manage the target problem earlier. However, they have tried to improve communication between the stakeholders of the company. The key representatives valued the target problem as high because it has a severe financial impact. It follows that the projects are not finalized on time. They said that the main causes of the target problem are lack of communication between the stakeholders and the way the company is dividing resources between the issues. Usually, an issue with fairly low priority does not get enough resources. They concluded that preventing the target problem is not an easy task. This would require increasing face-to-face meetings, increasing the number of inspections, and allocating skilled project managers to be responsible for the issues.

5. Results

In this section, we present the empirical results of the field studies. Section 5.1 presents the effort used of the cases. Section 5.2 presents the output of the ARCA method, and section 5.3 presents the feedback collected from the key representatives and case participants.

5.1. Effort used

Table 4 presents the effort used and the number of case participants throughout the different steps of the ARCA method. In total, 73–98 man-hours were required to conduct the cases. The required hours were mostly dependent on the number of case participants because both workshop sessions were time-boxed. The effort used increases with each additional case participant.

Roughly a quarter of the total effort was used in RCA facilitator-specific activities, whereas the rest was used in activities that included the case participants (see Table 4). An average of 10 h were used in step 1 (problem detection), 37 h were used in step 2 (root cause detection), 25 h were used in step 3 (corrective action innovation), and 12 h were used in step 4 (documentation of the results).

5.2. Output of the method

Table 5 presents the results of the method in the cases. The target problem causes were detected by the preliminary cause collection (52–108 causes) and by the causal analysis workshop (80–137 causes). The effort used was not fixed in the preliminary cause collection, whereas it was fixed in the causal analysis workshop.

It seems that the number of detected causes in the preliminary cause collection was dependent on the effort used. The correlation between the effort used and the number of problem causes in the preliminary cause collection is positive. It is also larger than the correlation between the number of case participants and the number of problem causes in the preliminary cause collection.

Our results indicate that a decreasing number of case participants detected an increasing number of causes in the causal analysis workshop. The duration of the workshop was fixed and the correlation between the effort used and the number of problem causes in the causal analysis workshop is negative.

A total of 2–6 root causes were selected in the cases. Together with their sub-root causes, the selected root causes formed a set of root causes that was processed in the corrective action workshop. In each case, 24–77 root causes were processed and 13–40 corrective actions were developed. The processed root causes covered 10–45% of the total number of the detected target problem causes in each case (average = 25%).

In case 2, the corrective action innovation step differed from those in the other cases. The corrective actions were developed by brainstorming each corrective action until a mutual understanding was found between the case participants. Thereafter, the next corrective action was developed, etc. All the other cases followed the brainwriting method, as presented in Section 3.2.3. This modification in case 2 (choosing to brainstorm instead of brainwrite) was done because we wanted to test whether brainstorming or brainwriting would better fit our needs. By comparing the number of corrective actions between the cases, case 2 was determined to be less effective than the other cases (see Table 5). Additionally, the quality of the corrective actions was lowest in case 2 (see Fig. 3), as their feasibility was relatively low. Based on our observations, the brainstorming method was less effective than the brainwriting method because the people were not able

Table 4
Effort used in the cases (h = hours) and the number of case participants (n) (* = RCA facilitator only).

The step of the ARCA method		Case 1		Case 2		Case 3		Case 4		Avg.		Std	
		h	n	h	n	h	n	h	n	h	n	h	n
Step 1	Problem definition meetings (startup)	17	10	10	5	6	6	6	4	9.6	6.3	5.3	2.6
Step 2	Preliminary cause collection (email inquiry)	3	7	5	5	3	6	1	4	3.2	5.5	1.5	1.3
	Organizing the cause-effect diagram (*)	9	1	10	1	17	2	9	1	11.3	1.3	3.9	0.5
	Causal analysis workshop	21	10	20	10	22	8	14	7	19.3	8.8	3.6	1.5
	Smartening up the cause-effect diagram (*)	4	1	4	1	4	1	4	1	4.0	1.0	0	0
Step 3	Root cause selection	6	5	6	8	3	6	5	7	5.2	6.5	1.5	1.3
	Corrective action workshop	23	8	24	11	18	8	16	7	20.3	8.5	3.9	1.7
Step 4	Final report (*)	12	1	12	1	12	1	12	1	12.0	1.0	0	0
Total (h)		98		96		90		73		89.3		11.4	

Please cite this article in press as: T.O.A. Lehtinen et al., Development and evaluation of a lightweight root cause analysis method (ARCA method) – Field studies at four software companies, Inform. Softw. Technol. (2011), doi:10.1016/j.infsof.2011.05.005

Table 5
Results of the method.

		Case 1	Case 2	Case 3	Case 4	Avg.	Std
Step 2	Target problem causes from the preliminary cause collection	93	108	66	52	80	25.4
	Target problem causes from the causal analysis workshop	80	137	105	116	110	23.7
Step 3	The number of selected root causes	6	2	5	6	5	1.9
	The number of processed root causes, including sub-root causes	41	24	77	42	46	22.3
	The number of corrective actions	38	13	33	40	31	12.4

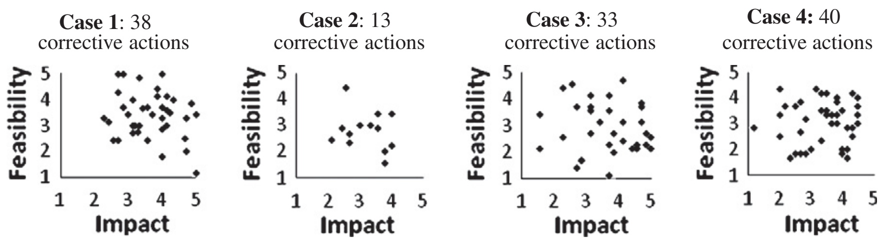

Fig. 3. Corrective actions of the cases (scales: 1 = low; 2, 3, 4, 5 = high).

to speak simultaneously. However, case 2 also varied from the other cases with respect to the homogeneity of the case participants (see Table 3). Thus, perhaps some important viewpoint was missing in case 2 when developing the corrective actions resulting in more unfeasible results. We do not know whether this was caused only by the brainstorming method or the method and the other case settings simultaneously.

High-quality corrective action is highly feasible and equally effective. Fig. 3 presents the impact and feasibility of the corrective actions per case as a scatter chart. In each case, every case participant evaluated the impact and feasibility of each corrective action to detect the highest-quality corrective actions, as presented in Section 3.2.3. The evaluations were done using a numerical scale, comprised of integers between one and five. We calculated the averages of the evaluations for each corrective action. The corrective action that had the highest value of the multiplication between the average impact and the average feasibility was interpreted as the highest-quality corrective action.

It is interesting that the proportion of the high-impact (avg. $\geqslant$ 3) corrective actions was larger than the proportion of the low-impact (avg. < 3) corrective actions in each case. Instead, the proportion of the high-feasibility (avg. $\geqslant$ 3) corrective actions was larger than the proportion of the low-feasibility (avg. < 3) corrective actions only in cases 1 and 4. It seems to be easier to develop high-impact corrective actions than to make them feasible.

5.3. Feedback of the case attendees

This section presents the feedback of the case participants and key representatives. Table 6 summarizes the data from the query forms after the causal analysis and corrective action workshops. There, the steps of the root cause detection and the corrective action innovation are presented from three different perspectives. The first perspective is the easiness of the method. The second perspective is the usefulness of the method. The third perspective emphasizes the quality of the outputs of the ARCA method, including the comparison of the method to the current process improvement practices of the case companies.

The results of the easiness and usefulness of the root cause detection step are combinations of multiple questions of the query forms (see Appendices C and D). The easiness of the root cause detection step is a combination of the factors "easiness of organizing causes" and "easiness of detecting root causes." The usefulness of the root cause detection is a combination of the factors "usefulness of the cause collection" and "usefulness of the method of root cause detection." The other results were rated with one question.

The case participants experienced the corrective action innovation step as highly easy to use (avg. = 5.9), whereas the step of the root cause detection was experienced as only slightly easy to use (avg. = 4.7). The participants experienced that both of these steps are useful. They also experienced that correct target problem

Table 6
Feedback of the case participants (*N* = the number of respondents, Avg. = average, Std = standard deviation, scale: 1 = very low; 2, 3, 4 = neutral; 5, 6, 7 = very high).

	Case 1			Case 2			Case 3			Case 4			All cases		
	N	Avg.	Std	*N*	Avg.	Std	*N*	Avg.	Std	*N*	Avg.	Std	*N*	Avg.	Std
Root cause detection															
Easiness	9	4.3	0.8	7	4.9	1.2	6	5.1	1.2	6	4.8	0.4	28	4.7	1.0
Usefulness	9	5.4	0.4	8	5.8	0.6	6	5.8	0.5	6	5.3	1.0	29	5.6	0.6
Correctness of detected causes	8	6.0	0.5	8	5.8	0.7	6	6.2	0.8	6	5.5	0.8	28	5.9	0.7
Openness in communication	9	5.9	0.9	9	6.2	0.7	6	6.7	0.8	6	6.2	0.4	30	6.2	0.8
Efficiency comparison to company practices	9	5.4	1.2	8	5.1	1.0	6	5.3	1.2	6	4.8	1.3	29	5.2	1.1
Corrective action innovation															
Easiness	7	5.7	1.0	10	6.0	0.8	7	6.0	0.6	6	6.0	0.6	30	5.9	0.7
Usefulness	6	4.8	1.0	8	5.0	1.1	7	5.1	1.1	6	5.0	0.6	27	5.0	0.9
Impact of the CAs	7	5.6	0.5	9	5.4	0.7	7	5.9	0.7	6	5.3	0.8	29	5.6	0.7
Feasibility of the CAs	7	5.3	0.5	10	4.4	1.1	7	5.3	0.8	6	5.7	0.8	30	5.1	0.9
Importance of processed causes for target problem	–	–	–	–	–	–	7	5.7	0.8	6	5.3	0.8	13	5.5	0.8
Importance of processed causes for product quality	7	5.4	0.5	10	5.6	0.8	7	6.3	0.8	6	5.3	0.5	30	5.7	0.8
Openness in communication	6	6.5	0.5	9	6.1	1.3	7	6.1	0.9	6	6.3	1.2	28	6.3	1.0
Efficiency comparison to company practices	6	6.2	1.0	8	6.0	0.9	7	6.1	0.7	6	6.3	0.5	27	6.1	0.8

Please cite this article in press as: T.O.A. Lehtinen et al., Development and evaluation of a lightweight root cause analysis method (ARCA method) – Field studies at four software companies, Inform. Softw. Technol. (2011), doi:10.1016/j.infsof.2011.05.005

causes were detected and that fairly feasible corrective actions that have a high impact on the target problem were developed. The communication in both steps was experienced as highly explicit. The impact of the corrective actions was evaluated to be generally higher than their feasibility. The case participants experienced that the processed root causes were important for both product quality and the target problem. Unfortunately, the evaluation of the importance of the processed root causes for the target problem was done only in cases 3 and 4. The case participants experienced the root cause detection step as a more effective method to detect new process improvement opportunities than their current process improvement practices (avg. = 5.2). Similarly, the corrective action innovation step was experienced as a more effective method to develop process improvement ideas (avg. 6.1).

Table 7 summarizes the answers of the key representatives when they were interviewed after the cases. Our goal was to evaluate how they experienced the easiness and usefulness of the ARCA method and to include the effort used with respect to the output of the method under the evaluation.

In general, it seems that the method was experienced as easy to use. On the other hand, organizing the causes was noted to be challenging (person 3b) and the assistance of the researchers made the method unnaturally easy to use (person 4). The key representatives' unanimous opinion was that their companies should adopt the method and that the results were experienced as beneficial in contrast to the effort used. Additionally, they were not able to name any other method that could reach equally advantageous results with lower costs than our RCA method. They experienced that significant root causes were detected with respect to the target problem, and most of them stressed that, if implemented, the developed corrective actions would have a high impact in preventing the target problems. As an exception, it was noted that the corrective actions do not prevent the target problem, but they do help the company to improve their processes (person 2).

6. Discussion

In this section, we answer the research questions and discuss our findings and possible threats to the validity of this study. In Section 6.1, we discuss the easiness and efficiency of the ARCA method in contrast to the current software process improvement practices of the case companies. In Section 6.2, we discuss the results of prior RCA studies and the feasibility of the ARCA method in contrast to the prior RCA methods introduced in Section 3.2. In Section 6.3, we discuss the validity of the conclusions based on the empirical results of this study.

6.1. Answering our research questions

One of our goals was to evaluate the ARCA method by answering the following research questions: "Is the ARCA method efficient?" and "Is the ARCA method easy to use?" Here, we answer these questions by discussing how the case attendees evaluated the usefulness and easiness of the ARCA method and the quality of its output.

Our results indicate that the effort required to use the ARCA method in similar case contexts is suitable. In Section 5.1, we showed that a total of 73–98 man-hours were required to conduct the cases with 7–11 case attendees. The key representatives experienced that the effort used was suitable in terms of the output of the method, as presented in Table 7. Furthermore, the case participants experienced the method as useful (see Table 6). Additionally, they experienced that the method is a more efficient practice to detect new process improvement opportunities and to develop process improvement ideas than their current company practices (see Table 6). Respectively, the key representatives were unable to name any method as efficient as the ARCA method (see Table 7). This evaluation logically covered the current process improvement practices of the case companies.

Hundreds of target problem causes were detected in the cases (see Table 5). The case participants experienced that the detected causes were correct (see Table 6) and the key representatives experienced that significant root causes were detected with respect to the target problems (see Table 7). These indicate that genuine and accurate target problem causes were detected. Our observations during the causal analysis workshops support this conclusion. In addition, the case participants experienced that the communication was highly explicit in the steps of the ARCA method (see Table 6).

Many high-quality corrective actions were developed for the processed root causes (see Fig. 3). The processed root causes were experienced as highly important for the target problem and

Table 7
Interviews of the key representatives. (Coding themes: E = method easiness, U = method usefulness, Q = output quality).

Question	Case 1	Case 2	Case 3		Case 4
	Person 1	Person 2	Person 3a	Person 3b	Person 4
How easy and learnable is the method?	"*Easy to use and internalize.*" (*E*)	"*Easy in contrast to required effort and the output of the method*" (*E, U, Q*)	"*Easy to use and learn*" (*E*)	"*It is fairly easy to use and learn. Organizing the causes was challenging*" (*E*)	"*It was easy with the assistance of the researchers*" (*E*)
Were the detected root causes significant with respect to the target problem?	"*Most of the causes were significant*" (Q)	"*As a general rule, yes. We have already reacted in one of the causes*" (Q)	"*Yes, they were. They matched well with my conception*" (Q)	"*Yes they were. I already knew some of those*" (Q)	"*Yes they were. The causes were mainly issues that lead the problem*" (Q)
Do the corrective actions prevent the target problem?	"*Yes, I think they do because they have a major impact on the processed root causes*" (Q)	"*No, I think that the corrective actions don't prevent the problem, but they do help us to improve our processes*" (Q)	"*Yes they do. We wouldn't even need to implement them all*" (Q)	"*I think that the corrective actions won't remove the problem completely, but they do have a major impact on the problem's sub-fields*" (Q)	"*Yes, the impact would be enormous*" (Q)
Would it have been possible to get the same results at lower costs using some other method?	"*No. We wouldn't be able to get this many relevant corrective actions*" (*U, Q*)	"*The method didn't require much effort. However, there should be only one workshop session and I would drop the email inquiry*" (*U*)	"*I don't believe that. I don't know any such method*" (*U*)	"*I think that 'better practice' would mean smaller group size and more talented experts in the second workshop*" (*U*)	"*Maybe some other brainwriting method, where ideas are developed in literal form, could work as well*" (*U*)
Should your company adopt the method?	"*Yes, we should. This works*" (*U*)	"*Maybe, because this is an easy method with much potential. Additionally, the costs are low*" (*E, U*)	"*I think we should adopt this method*" (*U*)	"*I would gladly try this method again. Formal prioritization was nice*" (*U*)	"*We should use this method, or at least a very similar one*" (*U*)

Please cite this article in press as: T.O.A. Lehtinen et al., Development and evaluation of a lightweight root cause analysis method (ARCA method) – Field studies at four software companies, Inform. Softw. Technol. (2011), doi:10.1016/j.infsof.2011.05.005

product quality (see Table 6), covering an average of 25% of the detected causes (see Table 5). The case participants experienced that feasible corrective actions that have a high impact on the target problem were developed (see Fig. 3, Table 6), and the key representatives stressed that the impact of the corrective actions in preventing the target problem would be high (see Table 7). These results indicate that important target problem causes were processed and high-quality corrective actions were developed to prevent them.

The results in Section 5.3 showed that, in general, the key representatives and the case participants experienced the ARCA method also as easy to use (see Tables 6 and 7). However, in each case, it was challenging to get a clear overview of the cause-effect diagram due to its enormous size. Therefore, it was also challenging to detect all the different effects to which a given cause was related. It was not a surprise that case participants evaluated the easiness of the root cause detection step with lower scores (avg. = 4.7) than the corrective action innovation step (avg. = 5.9). Maybe it is that organizing hundreds of target problem causes is more challenging than listing dozens of new ideas. It is likely that the case participants were also more familiar with the corrective action development practices, whereas analyzing the target problem causes systematically was something new for them.

In contrast to the prior process improvement practices of the case companies, we believe that the ARCA method is an efficient method to detect new software process improvement opportunities and to develop process improvement ideas. Our results additionally indicate that the ARCA method is relatively easy to use and learn.

6.2. Comparison to prior works

Some of our results follow the prior RCA studies. Grady [8] indicates that 7 h of team work is the minimum cost of conducting a non-recurring RCA method, whereas Mayes [6] indicates that the required effort to conduct an RCA method consists of 4–7 developers participating in a kickoff and a causal analysis meeting, each lasting 2 h, and 8–10 action team members using 10% of their time for action team duties.

Considering the target problem causes and the impact of the corrective actions, Card [15] discusses an RCA case where a total of 100 target problem causes were detected. There, the cause collection was conducted in a meeting to a certain extent similar to the causal analysis workshop of the ARCA method (see Section 3.2.2) resulting in an average of 110 target problem causes (see Table 5). Card [15] also presents quantitative evidence on the impact of the corrective actions developed through an RCA method in two software organizations. He claims that, when the DCA method (see Table 1) was used to prevent software defects, the impact of the corrective actions was enormous, resulting in a 50% decrease in the defect rates [15]. This indicates that focusing the software process improvement effort on the target problem causes probably decreases the likelihood of the target problem reoccurrence and, thus, slightly supports the evaluations of the case attendees on the impact of the corrective actions developed in our cases.

We noted that organizing the target problem causes is challenging. Other studies have faced similar problems. Usually, too many target problem causes are detected [5] and, overall, the causal analysis mechanism is qualitative and labor-intensive [8]. We believe it is important to use such a cause-effect diagram that makes organizing the target problem causes as easy as possible. Using a directed graph is currently a good candidate for this [4].

The prior RCA methods would have been less feasible in the cases of our field studies than the ARCA method. The DCA [15] and Proact RCA [3] methods are not as adaptable for various target problems as the ARCA method because they require accurate and reliable problem reports available for problem sampling, including a separate problem classification scheme for each target problem type [15]. Additionally, these RCA methods require heavy startup investments in problem classification scheme definitions, procedure setup, establishment of data collection mechanisms, and personnel training [15]. Our industrial partners would not have stood for such startup investments. The required startup effort of the ARCA method is relatively low, as it includes only the personnel learning. The RCA method presented by Rooney and Vanden Heuvel [2] would have required that potential problem causes are collected before the method can be even conducted. There obviously is a challenge in detecting the correct target problem causes in advance, as the target problems vary. Additionally, there would have been a problem of detecting too many problem causes [5], making the method highly difficult to use. We believe it is important to utilize both anonymous and public work practices when preventing cross-functional and complex target problems (see Section 2.2). This is not supported in any of the prior RCA methods. The PIC method [17] relies only on interviewing techniques, whereas the DCA [15] and Proact RCA [3] methods emphasize only meetings. In the prior RCA methods, there is also a problem of duplicating the same cause multiple times in the cause-effect diagram. Using a fishbone diagram [15], a logic tree [3], a list [17], a worksheet [17], or a chart [17] does not support references between the target problem causes, whereas the directed graph of the ARCA method (see Fig. 2) supports it. Finally, in the prior RCA methods, there is very little practical guidance on how to develop corrective actions.

6.3. Evaluation of the research

This section discusses the validity of our empirical results using a validation scheme presented by [35]. We will present the construct validity in Section 6.3.1, the external validity in Section 6.3.2, and the reliability of the study in Section 6.3.3. It should be mentioned that there is a fourth aspect of validity, called internal validity. However, even though it represents an important aspect, it is of concern only when the causal relations of the measured factors are examined [35]. Thus, this aspect is excluded here.

6.3.1. Construct validity

Construct validity reflects the extent to which the studied operational measures really represent what is investigated according to the research questions [35]. In this study, these are the measurements, query forms, and interviews that were carried out to evaluate the ARCA method.

We believe that high-quality corrective action has a high impact on the target problem, but, simultaneously, it is highly feasible. There is a threat to the construct validity regarding the evaluations of the quality of the corrective actions developed in the cases. The analyses were based on experiential evaluation of the case attendees only, not on monitoring the target problems afterward. Therefore, we do not know how many of the corrective actions were actually implemented, nor whether or not they had an impact in deterring the reoccurrence of the target problems. Generally, it should be noted that this sort of validity problem is common, as it is practically impossible to separate the effects of the RCA method from the company-specific context factors.

As the analyses of the impact and feasibility of the corrective actions are on uncertain ground, similarly is the conclusion on the suitability of the effort used. It is challenging to estimate whether the effort used was suitable or excessive, as there was no real evidence on either the costs required to implement the corrective actions or on their impact. As the only source was the opinions of the case attendees, the analyses of the usefulness of the ARCA method

Please cite this article in press as: T.O.A. Lehtinen et al., Development and evaluation of a lightweight root cause analysis method (ARCA method) – Field studies at four software companies, Inform. Softw. Technol. (2011), doi:10.1016/j.infsof.2011.05.005

(see Table 6) and suitability of the effort used with respect to the output of the method (see Table 7) are unreliable.

We compared the efficiency of the ARCA method to the current process improvement practices of the case companies, as the case attendees were asked to evaluate that. Unfortunately, we did not do such a comparison with the easiness of the ARCA method and, thus, the related conclusions are based solely on the personal experiences of the case attendees. Additionally, it is highly possible that some of the case participants were not experienced enough with the current process improvement practices of the case companies and, thus, their answers skewed our results. Fortunately, the key representatives were competent to perform such an evaluation, which increases the validity of the results. In addition, they performed this evaluation not only in the query forms but also in interview 2.

6.3.2. External validity

External validity is concerned with whether it is possible to generalize the findings of the study and to what extent they can be generalized [35]. In this study, this means that are our results also valid in other case contexts.

All of the cases varied and, thus, considered the ARCA method from different perspectives. Though the cases were conducted at four different companies, all with different case attendees and target problems, and though the interviews slightly differed between the cases, the results collectively confirmed the suitability of the ARCA method for medium-sized software companies where prior experiences with RCA are relatively insignificant. We believe that the results of this study can be generalized for similar case contexts.

The lack of comparison to prior RCA methods creates a severe threat to external validity. So far, we cannot conclude whether or not the ARCA method is truly efficient and easy to use in contrast to the prior RCA methods, as we were not able to compare its results extensively to those methods. The main cause for this is that no such prior results are publicly available and our field studies did not cover a case where the case attendees are highly experienced with prior RCA methods. We did not pilot those methods, either. We did an analytical argumentation on the selections we made during the ARCA method design and, when possible, we presented similar results from the prior works. However, our conclusions based on these are likely incomprehensive and inaccurate.

6.3.3. Reliability

Reliability is concerned with the extent to which data and analysis are dependent on a specific researcher [35]. Considering the reliability of our results, the fact that the researchers steered the cases with the key representatives (see Section 4.1.4) was both a strength and a weakness. The strength was that it made the cases more comparable, as almost everything was done similarly in the cases. On the other hand, the weakness was that the collected research data was partially bounded by the researchers' contributions. If the company people had tried to apply the method based only on the written instructions (see Sections 3.2.1–3.2.4), the evaluations of the effectiveness and easiness of the method could have been entirely different. We believe that the experience of the RCA facilitator has a great impact on the ARCA method output. Additionally, as we were a third party from the case attendees' point of view, it is possible that they were more or less willing to contribute in the cases than if the cases had been steered only by the company personnel. It is also possible that the high motivation and personal characteristics of the researchers spread to the participants, which had an impact to their motivation and open communication in the cases.

As the total number of the case attendees was only 30, the conclusions based on their feedback are on unreliable ground. Thus, our results should not be used to seek significant correlations between the work phases of the ARCA method and the feedback of the case attendees. Additionally, the small number of interviewees and cases likely skewed the interpretation of the results.

7. Conclusions and future work

It is argued that the key for effective problem prevention is to know why a problem occurs [2]. Unfortunately, in software engineering, there is very little practical knowledge on how the problem causes can be detected and prevented and what that requires. Our goal was to develop a lightweight RCA method and evaluate it through industrial field studies to introduce how problem causes can be detected and how the related corrective actions can be developed, as well as how much effort the RCA method requires and how the case attendees experience it.

This paper makes three contributions. First, we developed and introduced a lightweight RCA method, named the ARCA method. The ARCA method consists of four steps, i.e., target problem detection, root cause detection, corrective action innovation, and documentation of the results. Unlike the prior RCA methods applied in the software industry [5,8,13–15], the ARCA method does not require heavy startup investments and problem reports to detect its target problem. Instead, our method utilizes a focus group meeting to detect the target problem, making the method simultaneously highly adaptable for various target problems.

Second, we applied the ARCA method at medium-sized software product companies. This differs from the prior RCA studies that have investigated the use of RCA methods in large-company contexts [5,8,13–15] or student experiments [4]. In small and medium-sized software companies, the RCA method needs to be lightweight, as there is little possibility for separate resources for the RCA investigation, while there are more possibilities in a large company [6,8]. We also see that applying RCA to real industrial problems rather than the "toy" problems that are often used in student experiments consolidates the ARCA method in its true context.

Third, we provided empirical results of the usefulness, easiness, and output quality of the ARCA method, including the effort used in the cases. In prior works, such data is often missing. For example, in [15], the costs of the RCA method are reported only as a percentage of the yearly development budget instead of more concrete man-hours, as we did. Furthermore, the general satisfaction of the case attendees is not reported in any of the prior studies. We did that using interviews and query forms.

In contrast to the current process improvement practices of the case companies, the ARCA method was experienced as efficient. The effort of applying the method (89 man-hours, on average) was concluded to be suitable considering the value of the results. We showed in Sections 5.2 and 5.3 that the developed corrective actions were evaluated as fairly feasible and effective, having a high impact on the target problems. The case participants experienced that the steps of the root cause detection and corrective action innovation are both useful, and the key representatives experienced that it would not have been possible to get the same results with lower costs using any other method they knew. The method was generally experienced as easy to use. However, as an exception, organizing the detected causes was experienced as challenging due to the high number of detected causes.

We collected 757 target problem causes and 124 related corrective actions using RCA in the cases of this study. Analyzing the similarities between the target problem causes is part of our future work. The similarities between the developed corrective actions should be analyzed, as well. These would better help us to understand how the software companies try to prevent their problems and what types of related root causes exist.

Please cite this article in press as: T.O.A. Lehtinen et al., Development and evaluation of a lightweight root cause analysis method (ARCA method) – Field studies at four software companies, Inform. Softw. Technol. (2011), doi:10.1016/j.infsof.2011.05.005

Finally, to increase the validity of the study, the ARCA method needs to be used in different types of contexts, e.g. in software companies with extensive experiences with prior RCA methods. This also means that software companies should adopt and apply the ARCA method repeatedly.

Appendix A. Questions asked in interview 1 (group interview)

Part 1

1. How many employees work in your company?
2. How is problem prevention organized in your company?
3. How much effort does your company expend on software process improvement (SPI)?
4. What are the stakeholders attending to SPI in your company?
5. How does your company try to avoid quality deviations?
6. How are quality deviations detected in your company?
7. Are quality deviations other than software defects recorded?
8. How does your company react to quality deviations?
9. Are the causes of the quality deviations detected?
10. If so, how it is conducted and how many people are included in the analysis?
11. And if so, what stakeholders are present (developer, testers, designers, sales)?

Part 2

1. How much effort do you think your company has used to prevent the target problem previously?
   – How it is done?
2. In an economic sense, how significant is the target problem for your company?
3. How complex is the target problem and how would you characterize it?

Appendix B. Questions asked in interview 2

Part 3

1. Would it be easier to detect the same causes just by listing them generally?
2. Were the detected root causes significant compared to the target problem?
3. Were major deficiencies detected or were they more minor problems?

Part 4

1. Would it have been possible to develop similar process improvement ideas without root cause detection just by innovating generally in how you could improve your activities?
2. Would it have been possible to get the same results at a lower cost using some other practice?
3. In general, do the corrective actions prevent the problem?
4. Are the corrective actions feasible?
5. What is the impact of the corrective actions for other problems in your company?

Part 5

1. How easy and learnable is the RCA method?
2. Compared to the effort used, how would you characterize the feasibility of the RCA method?
3. Should your company adopt the RCA method?
4. What are the most relevant challenges in the RCA method that make it unfeasible for your company?

Appendix C. Questions asked on feedback form 1

1. The target problem

Answer the questions by giving a value [1 = very low; 2, 3, 4 = neutral; 5, 6, 7 = very high] that corresponds to the question best.

– Impact of the target problem for the quality of the product
– Adverse effect of the target problem on my daily work
– Difficulty of preventing the target problem
– Effort the company used to try to prevent the target problem earlier
– Impact of the target problem for the end users of the product
– Impact of the target problem on customer relationships
– Internal impact of the target problem for the company
– My experience of the technical causes of the target problem
– My knowledge of the impact of the target problem for the end users of the product

2. The quality of the causes and root causes

Answer the questions by giving a value [1 = very bad; 2, 3, 4 = neutral; 5, 6, 7 = very good] that corresponds to the question best.

– Usefulness of the cause collection
– Usefulness of the method of root cause detection
– Easiness of detecting the root causes
– Ability of the method to detect new process improvement opportunities in contrast to the current state of the practices of your company
– Correctness of the detected causes
– Correctness of the detected root causes
– Easiness of solving the detected root causes
– Openness of the communication in this first workshop session

3. Your duty in your company:___________________

4. Select the roles that best describe your responsibility in the company:

– I am a manager
– I am a developer
– I am a tester
– I am a salesman
– I am a trader
– Something else: _______________________________

5. How would you improve the RCA method?

Appendix D. Questions asked on feedback form 2

1. How much time you used to propose and evaluate the root causes to be processed before this workshop session:____________________
2. Were the processed root causes the most important with respect to the target problem?
   (Select one of the following)
   – Absolutely YES
   – More than YES
   – Yes

– Neutral
– No
– More than NO
– Absolutely NO

3. Were the processed root causes the most important with respect to the quality of the product?
   – (Select one of the following)
   – Absolutely YES
   – More than YES
   – Yes
   – Neutral
   – No
   – More than NO
   – Absolutely NO

4. Were the processed root causes easy to eliminate? (Select one of the following)
   – Absolutely YES
   – More than YES
   – Yes
   – Neutral
   – No
   – More than NO
   – Absolutely NO

5. The method used to develop the corrective actions
   Answer the questions by giving a value [1 = very bad; 2, 3, 4 = neutral; 5, 6, 7 = very good] that corresponds to the question best.
   – Easiness of the corrective action development method
   – Feasibility of the corrective action innovation method
   – Ability of the method to develop process improvement ideas in contrast to the current state of the practices of your company
   – Impact of the corrective actions on the target problem
   – Feasibility of the corrective actions
   – If implemented, the impact of the corrective actions for your company, in general
   – Openness of the communication in this second workshop session

6. How would you improve the RCA method?

Appendix E. Template for the corrective actions

The problem: -
The root cause -
Idea's name: -

Description:

Optimistic comments:

Skeptical comments:

	1 = minor; 2, 3, 4, 5 = major impact	1 = low; 2, 3, 4, 5 = high feasibility
Name	Impact (1-5)	Feasibility (1-5)
Total:		

References

[1] M. Kalinowski, G.H. Travassos, D.N. Card, Towards a defect prevention based process improvement approach, in: Proceedings of the 34th EUROMICRO Conference on Software Engineering and Advanced Applications, Parma, Italy, 2008, pp. 199–206.
[2] J.J. Rooney, L.N. Vanden Heuvel, Root cause analysis for beginners, Quality Progress 37 (7) (2004) 45–53.
[3] R.J. Latino, K.C. Latino (Eds.), Root Cause Analysis: Improving Performance for Bottom-Line Results. 6000 Broken Sound Parkway NW, Suite 300 Boca Raton, CRC Press, FL, 2006.
[4] F.O. Björnson, A.I. Wang, E. Arisholm, Improving the effectiveness of root cause analysis in post mortem analysis: a controlled experiment, Information and Software Technology 51 (1) (2009) 150–161.
[5] P. Jalote, N. Agrawal, Using defect analysis feedback for improving quality and productivity in iterative software development, in: Proceedings of the

Information Science and Communications Technology (ICICT 2005), 2005, pp. 701–714.

[6] R.G. Mays, Applications of defect prevention in software development, IEEE Journal on Selected Areas in Communications 8 (1990) 164–168.

[7] S.O. Al-Mamory, H. Zhang, Intrusion detection alarms reduction using root cause analysis and clustering, Computer Communications 32 (2) (2009) 419–430.

[8] R.B. Grady, Software failure analysis for high-return process improvement decisions, Hewlett-Packard Journal 47 (4) (1996) 15–25.

[9] M. Siekkinen, G. Urvoy-Keller, E.W. Biersack, D. Collange, A root cause analysis toolkit for TCP, Computer Networks (2008) 1846–1858.

[10] A. Traeger, I. Deras, E. Zadok, DARC: Dynamic Analysis of Root Causes of Latency Distributions, SIGMETRICS '08, Annapolis, Maryland, USA, 2008, pp. 277–288.

[11] T. Stålhane, Root Cause Analysis and Gap Analysis – A Tale of Two Methods, EuroSPI 2004, Trondheim, Norway, 2004, pp. 150–160.

[12] I. Bhandari, M. Halliday, E. Tarver, D. Brown, J. Chaar, R. Chillarege, A case study of software process improvement during development, IEEE Transactions on Software Engineering 19 (12) (1993) 1157–1170.

[13] M. Leszak, D.E. Perry, D. Stoll, A case study in root cause defect analysis, in: Proceedings of the 2000 International Conference on Software Engineering, 2000, pp. 428–437.

[14] A. Gupta, J. Li, R. Conradi, H. Rönneberg, E. Landre, A case study comparing defect profiles of a reused framework and of applications reusing it, Empirical Software Engineering 14 (2) (2008) 227–255.

[15] D.N. Card, Learning from our mistakes with defect causal analysis, IEEE Software 15 (1) (1998) 56–63.

[16] B. Andersen, T. Fagerhaug (Eds.), Root Cause Analysis: Simplified Tools and Techniques. United States, Milwaukee 53203: Tony A. William American Society for Quality, Quality Press, 2006.

[17] M. Ammerman, The Root Cause Analysis Handbook: A Simplified Approach to Identifying, Correcting, and Reporting Workplace Errors. 444 Park Avenue South, Suite 604, Productivity Press, New York, NY 1016, USA, 1998.

[18] D.N. Card, Defect-causal analysis drives down error rates, Quality Time 10 (4) (1993) 98–99.

[19] I. Burnstein, Practical Software Testing, Springer Science + Business Media, New York, 2003.

[20] Z.X. Jin, J. Hajdukiewicz, G. Ho, D. Chan, Y. Kow, Using root cause data analysis for requirements and knowledge elicitation, in: International Conference on Engineering Psychology and Cognitive Ergonomics (HCII 2007), Berlin, Germany, 2007, pp. 79–88.

[21] A.D. Livingstone, G. Jackson, K. Priestley, Root Causes Analysis: Literature Review, Health & Safety Executive, Contract Research Report 325, 2001, pp. 1–53.

[22] A.R. Hevner, S.T. March, J. Park, S. Ram, Design science in information systems research, MIS Quarterly 28 (1) (2004) 75–105.

[23] S.T. March, G.F. Smith, Design and natural science research on information technology, Decision Support Systems (15) (1995) 251–266.

[24] S. Kavadias, S.C. Sommer, The effects of problem structure and team diversity on brainstorming effectiveness, Management Science 55 (2009) 1899–1913.

[25] J.J. Rooney, L.N. Vanden Hauvel, Collecting data for root cause analysis, Quality Progress 36 (11) (2003) 104.

[26] R.L. Glass, Project retrospectives, and why they never happen, IEEE Software 19 (2002) 111–112.

[27] A. Burr, M. Owen (Eds.), Statistical Methods for Software Quality: Using Metrics for Process Improvement, ITP A Division of International Thomson Publishing Inc, 1996.

[28] W.J. Stevenson (Ed.), Operations Management, McGraw-Hill/Irwin, New York, 2005.

[29] T.C. Lethbridge, S. Elliott Sim, J. Singer, Studying software engineers: data collection techniques for software field studies, Empirical Software Engineering 10 (2005) 311–341.

[30] S. Wagner, Defect classification and defect types revisited, in: Proceedings of the 2008 Workshop on Defects in Large Software Systems (DEFECTS '08), Seattle, Washington, USA, 2008, pp. 39–40.

[31] M.V. Mäntylä, J. Itkonen, J. Iivonen, Who tested my software? Testing as an organizationally cross-cutting activity, Software Quality Journal, submitted for publication.

[32] R. Chillarege, I.S. Bhandari, J.K. Chaar, M.J. Halliday, D.S. Moebus, B.K. Ray, M. Wong, Orthogonal defect classification – a concept for in-process measurements, IEEE Transactions on Software Engineering 18 (11) (1992) 943–956.

[33] S.W. Gursimran, C.C. Jeffrey, A systematic literature review to identify and classify software requirement errors, Information and Software Technology 51 (7) (2009) 1087–1109.

[34] R.K. Yin (Ed.), Case Study Research: Design and Methods, Sage Publications, United States of America, 1994.

[35] P. Runeson, M. Höst, Guidelines for conducting and reporting case study research in software engineering, Empirical Software Engineering (14) (2008) 131–164.

[36] W. Foddy (Ed.), Constructing Questions for Interviews and Questionnaires, Cambridge University Press, Hong Kong by Colorcraft, 1994.

Please cite this article in press as: T.O.A. Lehtinen et al., Development and evaluation of a lightweight root cause analysis method (ARCA method) – Field studies at four software companies, Inform. Softw. Technol. (2011), doi:10.1016/j.infsof.2011.05.005