OPEN ACCESS

Lehtinen, Timo O.A.; Mäntylä, Mika V.; Itkonen, Juha; Vanhanen, Jari

Diagrams or structural lists in software project retrospectives - An experimental comparison

CrossMark

Diagrams or structural lists in software project retrospectives – An experimental comparison

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

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

ABSTRACT

Root cause analysis (RCA) is a recommended practice in retrospectives and cause–effect diagram (CED) is a commonly recommended technique for RCA. Our objective is to evaluate whether CED improves the outcome and perceived utility of RCA. We conducted a controlled experiment with 11 student software project teams by using a single factor paired design resulting in a total of 22 experimental units. Two visualization techniques of underlying causes were compared: CED and a structural list of causes. We used the output of RCA, questionnaires, and group interviews to compare the two techniques. In our results, CED increased the total number of detected causes. CED also increased the links between causes, thus, suggesting more structured analysis of problems. Furthermore, the participants perceived that CED improved organizing and outlining the detected causes. The implication of our results is that using CED in the RCA of retrospectives is recommended, yet, not mandatory as the groups also performed well with the structural list. In addition to increased number of detected causes, CED is visually more attractive and preferred by retrospective participants, even though it is somewhat harder to read and requires specific software tools.

1. Introduction

In software project retrospectives, individuals work together in order to create an understanding of what worked well in the prior project, and what could be improved (Bjørnson et al., 2009). Root cause analysis (RCA) is used in software project retrospectives, which are recommended practice for example in the Scrum software development method (Schwaber and Sutherland, 2011). RCA helps in capturing the lessons learned from individuals (Lehtinen et al., 2011) and aims to state what the perceived problem causes are and where they occur (Lehtinen and Mäntylä, 2011; Lehtinen et al., 2014a). Furthermore, RCA can be a part of project retrospectives, but it can also be a part of continuous software process optimization as recommended by the CMMI model (Software Engineering Institute).

A cause–effect diagram (CED) is a commonly recommended technique for RCA (Anbari et al., 2008; Bjørnson et al., 2009; Dingsøyr, 2005; Lehtinen et al., 2011). The diagram is used to register and visualize the outcome of RCA, i.e., the underlying causes of the problem. Its objective is to ease the detection and communication of the underlying causes and their causal structures. However, there are no studies comparing the use of CED with the use of textual notations, which represent the most straightforward approach to documenting retrospectives as they require no special tools other than a standard text editor. The use of structural lists can be thought as a natural baseline for such textual notations, which graphical diagrams, such as the CED, should be compared with. In our previous work, we operated with software organizations that have used textual notations to document the retrospectives instead of CEDs (Lehtinen et al., 2011, 2014b). Thus, reporting and visualizing the causal structures of a problem do not necessarily require CED and the benefits of CED have not been investigated in previous work.

Our research problem is the following: *Is CED needed in the RCA of software project retrospectives, and if so, why?* We studied the research problem by organizing a controlled student experiment as a part of a software engineering capstone project course, where students conduct software projects in industrial like environment. We compared the outcome of RCA and the perceptions of the retrospective participants between a CED and a structural list technique.

The rest of the paper is structured as follows. Section 2 introduces the related work, which includes using RCA in the retrospectives of software projects. Additionally, we will present how the CED and structural list techniques can be used in RCA to visualize and organize the causes of problems. At the end of the section, gaps in the existing research are presented. Section 3 presents the research objectives, questions, and methods. We will also introduce the research context,

---

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

research hypotheses, the used retrospective method (Bjørnson et al., 2009) and the experiment design including the treatments, response variables, and controlling the undesired variation. Section 4 presents the study results. Furthermore, we will answer the research questions and discuss the validity threats in Section 5. Section 6 summarizes our findings and suggests future work on the topic.

2. Related work

We start this section by presenting the concept of RCA in software project retrospectives. Thereafter, in Section 2.2 we discuss the effect of external representation for learning, including an introduction to CED and its comparison with textual notation techniques used in RCA. In Section 2.3 we conclude the gaps in the research.

2.1. Root cause analysis of software project retrospectives

Software project retrospectives, also known as postmortems, are aimed to facilitate learning from the success and failure of past projects. They are commonly defined as reflective practices (Babb et al., 2014), "powerful tools for project teams to collectively identify communication gaps and practices to improve future projects" (Bjarnason et al., 2014). Birk et al. (2002) stated that software project retrospectives provide an "excellent method for knowledge management", due to the high feasibility for continuous improvement and corrective action development. The objective of retrospectives is to help individuals, teams, and organizations to learn from the past (Dybå et al., 2014). This objective is fulfilled by sharing the lessons learned on the successful and unsuccessful events (Collier et al., 1996) over the members of software project organization (Lehtinen et al., 2014b). Such knowledge sharing increases the organizational knowledge (Boh et al., 2007), which in turn, becomes useful for software process improvement activities.

Software project retrospectives take a project success or failure as an input and provide the lessons learned, and possible improvement ideas, as an output. Root cause analysis is used in software project retrospectives to detect the underlying causes of the success and failure. It also helps to express how the underlying causes are related to one another (Lehtinen et al., 2014a). Stålhane et al. (2003) presented that such an approach is feasible for software organizations, because it 1) improves the documentation of knowledge, 2) improves the development of improvement actions, and 3) provides a good starting point for systematic knowledge harvesting. Card (1998) showed significant evidence on the high efficiency of using RCA in software project retrospectives, i.e., a 50% decrease in the defect rates during the two years of observations. Our prior studies (Lehtinen et al., 2014b, 2011) showed that RCA is also perceived as cost-efficient and easy-to-use by the retrospective participants. Furthermore, in a retrospective study comparing the causes of software project failures and successes, Moløkken-Østvold and Jørgensen (2005) indicate that the underlying factors of the success and failure are actually mirroring one another. This means that the same factors appear both as success factors reflecting the "good" practices, and failure factors, when neglected or misapplied, reflecting opportunities for process improvement. Yet, the current literature focuses mainly on the problems, since those reveal more direct opportunities for process improvement.

Software project retrospectives typically follow two work phases. First, the team members list and select success factors and problems occurred during the project or milestone (Bjørnson et al., 2009). It is important to focus on actions that truly have occurred, otherwise the retrospective becomes "an emotional vending sessions" (Bjarnason et al., 2014). Thereafter, the selected findings are further analyzed by the team members using RCA (Bjørnson et al., 2009). The team members conduct RCA by constantly asking "why?" for every cause detected (Lehtinen et al., 2011), e.g., by using Five Whys technique

(Andersen and Fagerhaug, 2006). While the causes are detected, they are also organized into CED (Bjørnson et al., 2009), an external representation of the RCA outcome. The ultimate output of RCA is the causal structure of events explaining why they occurred (Lehtinen et al., 2014a; Stålhane et al., 2003).

Unfortunately, software project retrospectives are often neglected (Dybå et al., 2014). Glass (2002) explained that this is because of too busy software teams, lack of retrospective timing, and lack of methodological support. In prior studies, software project retrospectives have been introduced as synchronous face-to-face meetings (Dingsøyr et al., 2001; Dingsøyr, 2005), but today's company practices favor distributed settings (Terzakis, 2011). Similarly, even though the use of CED has been introduced as an important part of retrospectives (Bjørnson et al., 2009), the company practices seem to favor textual notations to visualize the retrospective findings (Lehtinen et al., 2011, 2014b). Software tool support for collaborative cause–effect diagramming is also widely missing (Lehtinen et al., 2014b) and therefore using CEDs in the distributed settings is practically challenging. Thus, in terms of the tool support for modern distributed software project retrospectives, we should also determine how to visualize the outcome of RCA.

2.2. The effect of external representation for learning

The prior studies indicate that the external representation of knowledge impacts to the learning efficiency (Mayer and Gallini, 1990; Ainsworth and Th Loizou, 2003) and software project retrospective outcome (Bjørnson et al., 2009). Externalizing the tacit knowledge of individuals becomes important in retrospectives, because it enables organizational learning (Dingsøyr, 2005). The external representation is needed in order to control the problems of human memory (Von Zedtwitz, 2002; Siau, 2004). The external representation affects to the learning efficiency of individuals through "self-explanation" (Ainsworth and Th Loizou, 2003). Vessey (1991) stated that "problem presentation" and "problem solving task" strive the individuals to create mental models of problems, important for problem solution. Self-explanation has been recognized as a key mechanism for learning from problems (Ainsworth and Th Loizou, 2003). It is about developing "deeper understanding of material" by explaining the material whilst studying it (Ainsworth and Th Loizou, 2003). Self-explanation occurs in software project retrospectives, especially when the participants consider the tacit shared knowledge of others and their own. They develop deeper understanding about the occurred events and their mutual role in the project.

Three key factors for an effective external representation have been introduced. These are "Search", "Recognition", and "Inference" (Larkin and Simon, 1987). The *Search* factor expresses how easily the registered information can be found from the external representation. The notations of "visual languages" have been compared with textual notations. The prior studies indicate that the information encoding techniques are different and human mind also processes the different types of encodings differently (Moody, 2009). This means that the external representation potentially affects to the retrospective outcome, learning efficiency, and perceptions of participants. For example, Larkin and Simon (1987) claimed that in comparison with textual notations a diagrammatic representation provides a "smooth traversal" between the pieces of knowledge, which is important for problem solving.

The *Recognition* factor considers human abilities to recognize the information from the external representation. The representation techniques differ in terms of the expertise that is required to interpret the registered information (Moody, 2009). The prior studies claim that, in comparison with textual notations, extra training could be needed to interpret informationally equivalent diagrammatic representation (Ottensooser et al., 2012; Moody, 2009; Larkin and Simon, 1987). This means that the retrospective outcome could suffer from
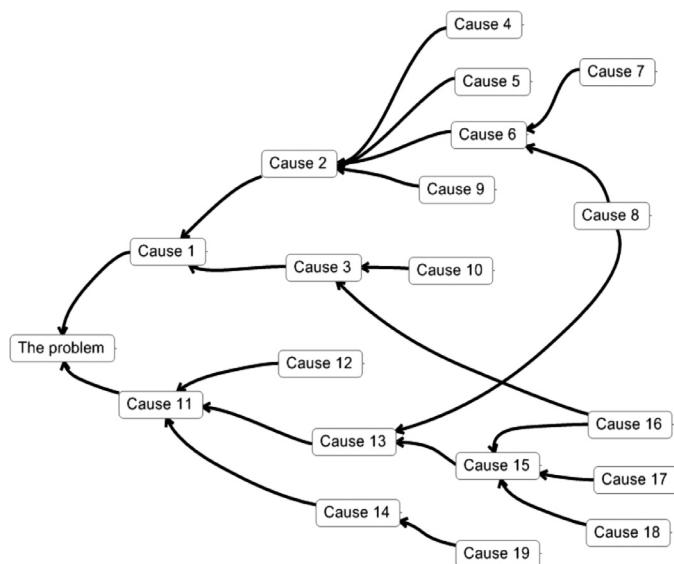
Fig. 1. The CED technique.

Fig. 2. The structural list technique.

lack of training. It follows that the retrospective participants remain unable to recognize the relevant information from the external representation (Larkin and Simon, 1987).

The *Inference* factor considers how to create linkages between the externally represented information in order to generate deeper level understanding on the underlying system of knowledge. Regarding the Inference, the prior studies indicate that an effective external representation presents a "cause-and-effect system", which helps the learner to create a "runnable mental model of the system" (Mayer and Gallini, 1990). The question is how to increase the efficiency of Inference with the external representation? Obviously, the individuals should be able to express cause–effect relationships over the separated pieces of information. Prior studies have claimed that a diagram representation increases the self-explanation efficiency (Ainsworth and Th Loizou, 2003) and learning efficiency (Mayer and Gallini, 1990). However, the effect for learning has been claimed to be valid only if the prior knowledge on the problem is low (Mayer and Gallini, 1990). In software project retrospectives, the participants teach and learn from one another, and they also generate new information by using self-explanation. Therefore, software project retrospectives could also benefit from the use of diagrams as the external representation technique.

Next, we present the related work of using CED and textual notation in project retrospectives, in Sections 2.2.1 and 2.2.2, respectively. Figs. 1 and 2 illustrate the differences between the two approaches.

2.2.1. The use of cause–effect diagrams in software project retrospectives

The use of diagram notations has been claimed to increase significantly the efficiency of self-explanation when compared with textual notations (Ainsworth and Th Loizou, 2003). In software project retrospectives, CEDs are the most frequently used techniques (Lehtinen et al., 2011). They are commonly used in RCA to register and visualize the causal structures of problems. Various techniques to draw CED are introduced, e.g., a fishbone diagram (Burnstein, 2003; Stevenson, 2005; Andersen and Fagerhaug, 2006; Ishikawa, 1990), a fault tree diagram (Andersen and Fagerhaug, 2006), a directed graph (Bjørnson et al., 2009), a matrix diagram (Nakashima et al., 1999), a scatter chart (Andersen and Fagerhaug, 2006), a logic tree (Latino and Latino, 2006), and a causal factor chart (Rooney and Vanden Heuvel, 2004). However, only few of them are utilized in software project retrospectives. These include the fishbone diagram (Burnstein, 2003; Andersen and Fagerhaug, 2006; Stevenson, 2005; Bjørnson, Wang, and Arisholm, 2009; Stålhane, 2004; Stålhane et al., 2003) and
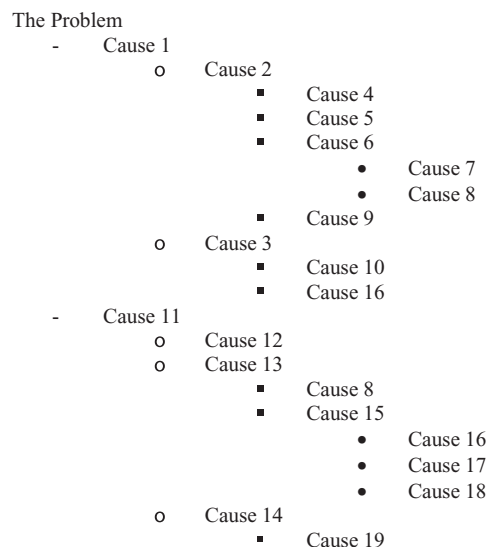
directed graph (Bjørnson et al., 2009; Lehtinen et al., 2011, 2014b). The fishbone diagram applies a tree structure where the causes of problems are organized into some premade classes of causes (Lehtinen et al., 2011). Instead, the directed graph applies a network structure where the causes of problems are organized solely based on their cause and effect relationships (Lehtinen et al., 2011). An example of directed graph structure is illustrated in Fig. 1.

Bjørnson et al. (2009) compared the use of the fishbone diagram with the directed graph in software project retrospectives. They found that the directed graph outperformed the fishbone diagram in the number of detected causes, which means that the outcome of RCA is dependent on the external representation technique used to visualize the causes. The comparison also revealed that the directed graph improves the analysis by increasing the number of hubs, which are defined as causes that are related to more than one problem (Bjørnson et al., 2009). The increasing number of hubs indicates improvement in the Inference factor (Mayer and Gallini, 1990). The strict hierarchical manner and weak layout of the fishbone diagram are its main weaknesses (Bjørnson et al., 2009). Another problem of the fishbone diagram is a tree structure (Lehtinen et al., 2011). The tree structure enforces duplicating the same cause under many problems whereas in the network structure only references to the problems are duplicated (Lehtinen et al., 2011). Thus, in the network structure, the number of cause statements remains as low as possible. The network structure also makes the linkages between the causes and problems visual, which associates with improvements in the self-explanation and Inference.

2.2.2. The use of structural list in software project retrospectives

A structural list is an alternative approach to CED. It is a textual representation used to register and visualize the cause–effect structures of problems. An example of a structural list is illustrated in Fig. 2. Ammerman (1998) presented a technique for RCA called Causal Factor List. He claims that listing the causes into a computer file helps in detecting the root causes of problems. Drawing CED requires writing down cause statements with graphical nodes and edges to interconnect the detected causes (Dingsøyr et al., 2001). Instead, listing the causes requires only that the cause statements are written down and simultaneously placed under one another. Additionally, making a structural list of causes does not require specific software tools for RCA as it is with CEDs (Lehtinen et al., 2011, 2014b).

Furthermore, the retrospective outcome and the perceptions of participants utilizing a structural list have rarely been compared with the use of CED (Stålhane, 2004; Stålhane et al., 2003). In our prior

study (Lehtinen et al., 2011), we criticized the feasibility of using the structural list technique in RCA. We assumed that in the context of software engineering, using that technique makes the analysis difficult, because of the high number of detected causes (Lehtinen et al., 2011). In addition, the structural list has the same practical problem as the fishbone diagram; when a cause explains more than one effect, you need to place the same cause under many effects. This means that when using the structural list in RCA, writing down the causes more than once increases the workload (Lehtinen et al., 2011). However, comparison between the fishbone diagram and the directed graph (Bjørnson et al., 2009) is not enough for determining the effectiveness of using the structural list, because the fishbone diagram utilizes different visual structure than the structural list.

2.3. Gap in the research

The prior studies on cognitive psychology and human factors (Ainsworth and Th Loizou, 2003; Larkin and Simon, 1987) indicate that use of diagrams could improve the efficiency of learning in software project retrospectives. However, the prior studies have not considered the effect of external representation for generating new information. Instead, they have only considered the learning efficiency from a premade knowledge, e.g., learning how the blood vessel is functioning (Ainsworth and Th Loizou, 2003).

The prior studies have also failed to address the questions whether the use of CED outperforms textual notations formulated as a structural list (Ammerman, 1998) during the RCA of retrospectives. Instead, the prior studies have indicated that the effectiveness of RCA is dependent on the technique used to visualize the causes of problems (Bjørnson et al., 2009; Lehtinen et al., 2011). Yet, those studies compare two different CED techniques rather than comparing them directly with the structural lists. Comparison to structural lists is important as they are the most straightforward to use and they are used in industry (Lehtinen et al., 2011, 2014b).

Making structural lists does not require drawing nodes and arrows between the causes of problems as it is with CEDs. Therefore, they neither require specific software tools (Lehtinen et al., 2011, 2014b). Thus, it is possible that a textual notation in the form of a structural list is a more effective technique than using CED. The results of Ottensooser et al. (2012) who compared the use of textual and graphical notations for interpreting business process descriptions support this idea. On the other hand, it is also possible that it is precisely the arrows and nodes of CEDs which improve the retrospective outcome and the perceptions of participants as they help to visualize and remember the causal structures of problems. The prior studies on organizational learning systems and "cognitive maps" support this view (Lee et al., 1992). Finally, the evaluation needs to be done in the actual software project retrospective context, because "different representations of information are suitable for different tasks and different audiences" (Moody, 2009).

3. Research methods

In this section, we introduce the research goals and present how the research data was collected and analyzed in this controlled experiment (Juristo and Moreno, 2003). Research objectives and questions are introduced in Section 3.1. Thereafter, the research context is presented in Section 3.2. In Section 3.3, we introduce the experimental design including the used retrospective method and the treatments, response variables and controlling the undesired variation. Section 3.4 introduces the data collection and analysis methods.

3.1. Research objectives and questions

Our objective is to compare two cause and effect structuring techniques used in software project retrospectives: 1) a directed graph (Bjørnson et al., 2009; Lehtinen et al., 2011), and 2) a structural list (Ammerman, 1998). The directed graph has been presented as the most optimal CED technique in the RCA of software project retrospectives (Bjørnson et al., 2009; Lehtinen et al., 2011).

We compare the outcome of RCA, i.e., the number and causal structures of the detected causes considering both the total number of causes and the number of causes with specific characteristics. We also compare the perceptions of the participants about the techniques. The research aims to answer the following comparative questions:

RQ1: Is there a difference between the techniques in terms of the outcome of RCA?

    RQ1a: *Is there a difference in the number of the detected causes?*

    RQ1b: *Is there a difference in the structures of the detected causes?*

    RQ1c: *Is there a difference in the characteristics of the detected causes?*

RQ2: Is there a difference between the techniques in terms of the perceptions of retrospective participants?

    RQ2a: *Is there a difference in the preferred technique?*

    RQ2b: *How do the retrospective participants evaluate and describe the techniques?*

3.2. Research context

Since the early 1980s, Aalto University has provided a capstone project course for computer science students (Vanhanen et al., 2012). During the course, the students develop software for external customers in teams. The software development for each customer is arranged as a software project lasting for five months. Each student uses approximately 150 h for the project. Based on our experiences and the course feedback, the students are highly committed to the projects. The project teams have a total of seven to nine student members. These include a project manager, a quality manager, a software architect and four to six developers. There are no freshmen students in the course. The managers are M.Sc. level students whereas the developers are B.Sc. level students. Many students already have years of experience on industrial software development.

The teams are required to follow a process framework defined by the course (Vanhanen et al., 2012). The process framework divides the projects into three timeboxed *iterations*, each lasting six to seven weeks. The process framework combines practices from both agile and plan-driven process models. These can be adapted to sprints, iteration planning, iteration demos, backlogs, weekly stand-ups, retrospectives, pair-programming, continuous integration, risk management, effort estimation and realization, use-cases, functional testing, and more rigorous quality assurance. Each team is responsible for planning and using a development process that follows the process framework.

The use of students as study subjects has been discussed in the software engineering literature (e.g., Svahnberg et al., 2008; Berander, 2004; Carver et al., 2003; Runeson, 2003; Höst et al., 2000). Runeson (2003) discussed the difference of using freshmen students, graduate level students, and industry personnel as study subjects. The conclusions are that graduate level students are feasible subjects for revealing improvement trends, but infeasible to reveal the absolute levels of improvements (Runeson, 2003). Berander (2004) explained that the applicability of using students as study subjects is dependent on their experience and commitment. He also claims that the use of students "as representatives for professionals" is more appropriate in software projects than classroom settings (Berander, 2004). Similar conclusions are also given by Carver et al. (2003).

The experiment was conducted in the retrospectives of 11 project teams out of 14 during the academic year 2010–2011. The

participation in the experiment was voluntary for the project teams. The team members did not know the objective of the experiment in advance. The research context was feasible for studying the improvement trend over the use of CED and structural list in the software project retrospectives of small teams. Most of the student subjects were graduate level students, who were experienced on software development and committed to their software projects. Thus, in the retrospectives, they were able to consider software project problems, which were relevant to their teams. The course projects were also similar to "real" projects and many challenges encountered by the student teams were industrially relevant. The challenges were mainly related to system functionality, system quality, communication, and taking responsibility. The detailed qualitative analysis of the causes is published in another paper (Vanhanen and Lehtinen, 2014). The customers were also committed to their projects and they paid a fee for the university when they got a student project. Thus, the students were required to develop software that was truly needed by the customers. Additionally, similar research context has been previously used to conduct somewhat similar comparison (Bjørnson et al., 2009).

3.3. Experiment design

For the participating project teams (see Section 3.2), we provided the retrospective methodologies and controlled the retrospective settings. The course framework required the teams to conduct a retrospective at the end of the second and third iteration. The retrospective method and the used effort were fixed (see Section 3.3.1). Thus, our design had two experimental units (retrospectives) for each participating project team, meaning 22 experimental units as a total.

The experiment followed a single factor paired design with a single blocking variable (Juristo and Moreno, 2003). The factor that we examined was the technique used to visualize and organize the causes of problems. The factor had two alternatives: CED and a structural list. Both of these treatments were applied by each team, but in different retrospectives starting in randomized order. Fig. 1 introduces the CED and Fig. 2 introduces the structural list technique. In CED, arrows are drawn between the causes of the problem. Instead, in the structural list, the causal structure is visualized using bullet lists. Furthermore, if a cause affects more than one effect, multiple arrows are drawn from the cause when using CED. Instead, with the structural list such cause needs to be duplicated under each effect it explains (see causes 8 and 16 in Figs. 1 and 2).

The blocking variable that we were not able to eliminate was the project phase where the retrospectives were conducted. The first retrospective was conducted in the middle (Iteration 2) and the second was conducted at the end of the project (Iteration 3). We balanced our experiment design in order to take the project phase into account

Table 1
Distribution of treatments (A ; CED, B ; the structural list) into 22 experimental units.

| | | Team (T) | | | | | | | | | | |
		T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11
Phase (I)	I2	A	A	B	A	A	A	B	B	B	A	B
	I3	B	B	A	B	B	B	A	A	A	B	A

in the analysis. Table 1 summarizes the experiment design including the distribution of teams in the treatments and the project phase. The starting order of treatments was randomized for each team. As a result, six teams used CED and five teams used the structural list in the first retrospective (Iteration 2). Respectively, six teams used the structural list and five teams used CED in the second retrospective (Iteration 3). This randomization balanced the potential effects of the blocking variable related to the project phase. Furthermore, our data analyses were conducted as a paired analysis comparing the differences of the treatments inside each team, which mitigates the effects of differences between teams.

3.3.1. Retrospective method

The used retrospective method, summarized in Fig. 3, started with a short introduction about the method. We presented for the participants how the steps of problem detection and root cause analysis will be conducted in the retrospective. Our method follows the post-mortem analysis method introduced by Bjørnson et al. (2009) who claimed that such a retrospective method is lightweight and feasible for small software project teams. The first author acted as the facilitator of the retrospectives. He introduced the problem detection and root cause analysis steps for the participants and thereafter acted as the scribe. The method consists of two separated steps, which are introduced below.

In the first step (problem detection), the participants were asked to write down problems, which have had a negative impact on reaching the project goals. Thereafter, each participant introduced the problems to the others. The facilitator registered the problems and projected them on the wall by the first author who acted as a scribe. Similar problems were grouped together by the participants. Thereafter, the participants voted two problems for RCA. These problems are referred to as *voted problems* later in this article. The first step was timeboxed to about 30 min.

The second step (root cause analysis) was conducted for both of the voted problems separately, lasting 40 min for each problem. First, each participant alone wrote down causes for the voted problem (5 min). Thereafter, they presented the causes for the others who simultaneously brainstormed more causes (15 min). The facilitator registered all detected causes immediately to a cause and effect
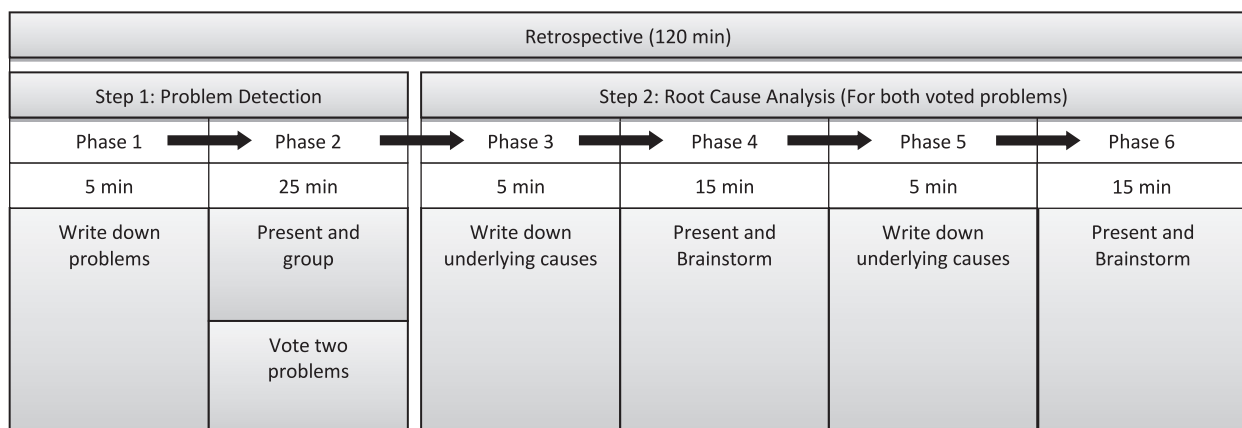

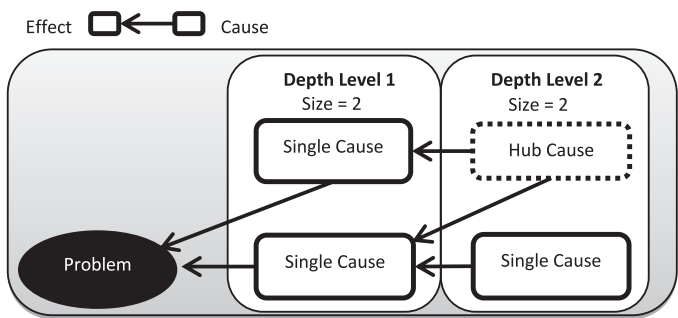**Fig. 3.** The retrospective method used in the study.

Fig. 4. Taxonomy used to clarify our research hypotheses.

structure shown on the wall. These two phases were repeated once more for the same voted problem. The second voted problem was thereafter processed.

3.3.2. Response variables and research hypothesis

Fig. 4 introduces the taxonomy used to clarify our research hypotheses. The figure draws a simple *causal structure* for a problem. The problem is placed on the left side of the figure while its causes are placed on the right side. The causes are organized based on their cause and effect relationships. Theoretically, each cause creates an *effect* (or *effects*), which itself can be a cause or the problem, and it is affected by its *sub-cause(s)*. In the figure, the causes being placed next to the problem are the effects of their sub-causes placed on the right side of the diagram. In order to simplify our terminology, each cause, effect and sub-cause explaining why the problem occurs is *a cause of the problem*.

Furthermore, *depth level of a cause* indicates the number of causes on the shortest path from the cause to the problem. Additionally, the size of a depth level (x) indicates the total number of causes having the depth level n. In Fig. 4, we can see that the size of the depth level (1) is 2. Finally, a hub cause (Bjørnson et al., 2009) refers to a cause that creates more than one effect and a single cause refers to a cause that creates exactly one effect.

Table 2 summarizes the response variables, our research hypotheses, and the measurements that we used. The response variable *cause count* (CC) is the number of problem causes detected in a retrospective. It indicates how actively the participants presented their visions about the software project, one of the key requirements for a successful retrospective meeting and organizational learning (Dingsøyr, 2005). It has been claimed that the number of detected causes also indicates the effectiveness of the RCA method (Bjørnson et al., 2009). However, measuring the effectiveness of the RCA method with the number of detected causes is somewhat an inappropriate approach, because the measurement does not say anything about the correctness and relevancy of the detected causes. CC is a simple indicator that counts the number of the detected causes while ignoring their actual content and related causal structures. For example, there are 19 causes in Figs. 1 and 2. Thus, the CC would be *19* for both figures. Our hypothesis was that the retrospective method utilizing CED re-

sults in a higher CC than the one utilizing the structural list. We based this hypothesis on prior studies that have commonly recommended using CEDs in RCA and also found it as a more efficient approach for learning than the structural list (see Section 2.2).

Causal structure indicates the cause and effect structure of the causes of the problem. We use two response variables related to the causal structure, proposed by Bjørnson et al. (2009), the *size of depth level* (SoDL) and the *proportion of hub causes* (PoH) (see Fig. 4). The function SoDL(x) indicates the number of causes being registered to the depth level x, whereas the PoH value indicates the proportion of detected causes which explain more than one effect. Our hypothesis was that generally the return value of SoDL(x) increases among the depth levels. This hypothesis was based on our prior experiences on the output of RCA in industrial software project context (Lehtinen and Mäntylä, 2011). In RCA, the detection of causes starts by the detection of few "first level causes" (Andersen and Fagerhaug, 2006), which thereafter evolve to the detection of "higher level causes" (Andersen and Fagerhaug, 2006) resulting in increasing number of detected problems and causes at the higher depth levels. We also hypothesized that the return value of SoDL(x) increases more with CED than with the structural list. This hypothesis was based on our understanding about the visual structure of CED. In contrast to the structural list, CED uses graphical nodes and edges (see Fig. 1) helping the participants to remember (Ainsworth and Th Loizou, 2003) and focus on (Larkin and Simon, 1987) the detected causes. Additionally, CED utilizes network structure which maintains the causal structure as clean and simple. Thus, we assumed that higher numbers of causes are detected at the higher depth levels when CED is used. The return value of SoDL(x) is measured by calculating the number of causes at the corresponding depth level x.

Furthermore, our hypothesis was that the PoH value is higher when CED is used. The prior studies support this hypothesis as they have indicated improvements in the self-explanation efficiency (Ainsworth and Th Loizou, 2003) and Inference (Larkin and Simon, 1987) while a diagram representation has been compared with a textual representation. In CED, arrows are drawn between the cause and its effects. Instead, in the structural list, the cause needs to be duplicated under the effects it explains. Thus, the number of cause statements is lower in CED than it is with the structural list. Additionally, unlike the structural list, the arrows between the causes and effects keep their relationships visible. There is simply less distraction in the causal structure when CED is used and the structure is also visual making it easier to remember (Ainsworth and Th Loizou, 2003). Thus, it is also likely easier to detect the different effects the cause explains. We think that the more there are hub causes, the more extensively the causal relationships are analyzed. This is because the hub causes create interconnections between larger ensembles of causes than interconnections between few individual causes. The PoH value is measured by calculating the percentage of causes that were used to explain more than one effect.

Characteristics of detected causes (CDC) indicate the distribution of the detected causes among process areas and cause types. Our hypothesis was that the CDC is not dependent on the treatments. We

Table 2
Response variables, research hypotheses, and related measurements used.

Response variable	Research hypothesis	Measurement
Cause count (CC)	CC with diagram > CC with list	The number of causes
Causal structure Size of depth levels (SoDL)	$SoDL(n+1) > SoDL(n) > \cdots > SoDL(2) > SoDL(1)$	The number of causes at different depth levels
	$\dfrac{SoDL(n+1)\,with\,\text{diagram}}{SoDL(n+1)\,with\,\text{list}} > 1$	The number of causes at different depth levels
Proportion of hub causes (PoH)	PoH with diagram > PoH with list	The percentage of causes that were used to explain more than one effect
Characteristics of detected causes (CDC)	CDC with diagram ≈ CDC with list	Distributions of classified causes
Perceptions of participants (PP)	PP with diagram > PP with list	Questionnaires and group interviews

based this hypothesis on the fact that neither of the treatments steers the participants to consider some specific project areas or cause types. We believed that the CDC was mostly dependent on the teams and problems analyzed, not on the studied techniques used to organize and visualize the problems and their causes. CDC is measured by using a classification system for the detected causes. We compared the distributions of causes in cause classes over the treatments.

Perceptions of participants (PP) reflect the evaluations of the participants on the treatments. Considering the PP, our initial hypothesis was that the participants prefer CED to be used in retrospectives. This hypothesis was based on prior studies that have commonly recommended using CEDs in RCA (see Section 2.2.1). We used a questionnaire (see Appendix A) after each retrospective to measure the perceptions of participants. Additionally, after both treatments were conducted, we used another questionnaire (see Appendix B) combined with a group interview in order to conclude which treatment the participants preferred and why.

3.3.3. Controlling undesired variation

We assumed that it was highly possible that the project phase where the retrospective was conducted had an impact on the retrospective outcome. We also assumed that the retrospective outcome is highly dependent on the team. In order to balance the effects of these variables, the treatment of each team was randomly assigned in the first phase. In addition, we applied both treatments to each team and used paired analysis to mitigate the variations between the teams.

We ensured that the retrospective settings were similar in each experimental unit. Therefore, six context variables were controlled. The context variables included the retrospective goal, the number and roles of the participants, the used language, the physical settings, and the retrospective facilitator. We also identified and measured three confounding variables, since we had no control organizing the teams and the project topics. The confounding variables included the voted problems, team members' motivation, and team spirit.

We controlled the goal of each retrospective. This was important as the problems related to software projects and the number and characteristics of their underlying causes vary (Lehtinen and Mäntylä, 2011). Thus, our study results were dependent on the problems analyzed. We controlled this issue by forcing each team to analyze a common endemic problem that occurs frequently during the projects, i.e. "*why it is challenging to reach the project goals*" (Vanhanen et al., 2012).

The number and roles of retrospective participants were controlled. This was important as we believe that the number and causal structures of the causes of a problem are dependent on the number of participants. A high deviation in the number of participants between the treatments would likely have biased the study results. We decided that each retrospective has to include at least four to seven participants, as suggested in Lehtinen et al. (2011) . Additionally, the maximum deviation in the number of participants between the two retrospectives of each team was limited to ±1. Similarly, the roles of the participants were controlled. It was decided that at least two out of three people in the management roles of the team have to be present at both retrospectives.

The used language was controlled. This was important as we believe that the team members' contribution is dependent on the language used. People are likely more active speakers when they use their own mother tongue and thus also the output of retrospectives is dependent on the language used. It was decided that the teams have to use the same language in both treatments.

Every retrospective was conducted in similar physical conditions. We took care that the infrastructure used to register and visualize the problems and their causes did not change between the retrospectives, i.e., the used laptop, software tools (Mindjet and MS Word) and projector. This was important as the screen resolution, margins, zoom level, etc. could have otherwise biased the study results through vary-

ing visualization capabilities. Similarly, the meeting room settings including the room size, lighting and location remained similar.

We also controlled the facilitator of the retrospectives. The first author of this paper steered each retrospective and acted as the scribe for each team. This was important as thus we were able to control the skills of the facilitator. The first author has prior experiences on steering RCA and he was also familiar with the used software tools.

Three confounding variables were measured in order to evaluate that dramatic changes in the working of the team did not happen between the retrospectives. The confounding variables included the voted problems (see Table 5), team members' motivation and team spirit. Considering the voted problems, we compared the problems the retrospective participants selected for RCA in each treatment. This was important as now we were able to evaluate whether the differences in the treatments may have been caused by different problems analyzed. Furthermore, considering the team members' motivation and team spirit, we used a questionnaire after each retrospective, as introduced in Section 3.4.3. This was also important as now we were able to evaluate whether the differences between the treatments were caused by varying motivation or team spirit. We asked the participants to evaluate their personal effort, their team's effort, the openness in communication, and the team spirit in each retrospective. We also asked them to evaluate 1) whether some participants purposefully left some important causes out of their attention and 2) whether the participants did not dare to name all the detected causes publicly.

3.4. Data collection and analysis

In this section, we introduce the methods we used in the data collection and analysis. As a summary, the data collection was based on triangulation which increases the validity of the study results (Yin, 1994; Runeson and Höst, 2008; Jick, 1979). We used the output of RCA in statistical analyses on the cause count and causal structures of the treatments (see Section 3.4.1). Additionally, we used the output of RCA to analyze whether the characteristics of detected causes remained similar over the treatments (see Section 3.4.2). Furthermore, we combined statistical methods with qualitative methods in order to evaluate the perceptions of participants about the treatments. We asked the participants to provide feedback by using questionnaires (see Section 3.4.3) and group interviews (see Section 3.4.4). Each retrospective and group interview was video recorded in order to be able to transcribe the interviews and further analyze the retrospectives if needed.

3.4.1. Cause count and causal structures

The cause count was analyzed with the paired-samples two-tailed *t*-test with the alpha level 0.05. We compared the number of detected causes in the retrospectives of each team. Each cause was counted only once, i.e., the duplicate cause statements were removed. As the number of retrospective participants varied ±1, we also compared the number of detected causes per number of participants. We also analyzed the cause count by comparing the average, minimum, lower quartile, median, upper quartile, and maximum number of detected causes between the treatments.

The causal structures were analyzed by comparing the size of depth levels, and the proportion of hub causes between the treatments. In the comparison, we used the paired-samples two-tailed *t*-test with the alpha level 0.05. Between the treatments of each team, we analyzed whether CED results systematically in larger sizes of depth levels than the structural list technique. Furthermore, we also analyzed whether CED systematically results in a larger proportion of hub causes.

Using the *t*-test was reasonable as the number of detected causes in the treatments was normally distributed between the teams. This conclusion was based on the Shapiro–Wilk test and the analysis of related Q–Q plots. We also tested that the distributions of causes at

Table 3
Process areas of the classification system express where the causes occur (Lehtinen and Mäntylä, 2011).

Process area	General characterization of the detected causes
Management work (MA)	Company support and the way the project stakeholders are managed and allocated to tasks.
Sales and requirements (S&R)	Requirements and input from customers.
Implementation work (IM)	The design and implementation of features including defect fixing.
Software testing (ST)	Test design, execution, and reporting.
Release and deployment (PD)	Releasing and deploying the product.
Unknown (UN)	Causes that cannot be focused on any specific process area.

Table 4
Cause types of the classification system express what the causes are (Lehtinen and Mäntylä, 2011).

Type/sub-type	General characterization of the detected causes
People (P)	This cause type includes the people related causes.
Instructions and experiences	Missing or inaccurate documentation and lack of individual experience.
Values and responsibilities	Bad attitude and lack of taking responsibility.
Cooperation	Inactive, inaccurate, or missing communication.
Company policies	Not following the company policies.
Tasks (T)	This cause type includes the task related causes.
Task output	Low quality task output.
Task difficulty	The task requires too much effort, or time, or it is highly challenging.
Task priority	Missing, wrong, or too low task priority.
Methods (M)	This cause type includes the methodological causes.
Work practices	Missing or inadequate work practices.
Process	The process model is missing, unclear, vague, too heavy, or inadequate.
Monitoring	Lack of monitoring.
Environment (E)	This cause type includes the environment related causes.
Existing product	Complex or badly implemented existing product.
Resources and schedules	Wrong resources and schedules.
Tools	Missing or insufficient tools.
Customers and users	Customers' and users' expectations and need.

depth levels were normally distributed. The number of causes was normally distributed from the first to sixth depth levels.

Furthermore, we evaluated the standardized effect size for the systematic differences between the treatments by using Cohen's d (1988). This was done by dividing the difference between the means of treatments with their pooled standard deviation. The effect size results were interpreted in the following way: $d < 0.2$ (small), $d \approx 0.5$ (medium), and $d > 0.8$ (large) (Cohen, 1988). The following pattern was used to calculate Cohen's d, where X is the sample mean, n_i is the sample size, and s_i is the standard deviation (Kampenes et al., 2007):

$$\text{Cohen's } d = \frac{X_1 - X_2}{\sqrt{\frac{(n_1 s_1{}^2 + n_2 s_2{}^2)}{(n_1 + n_2)}}}$$

3.4.2. Characteristics of detected causes

We evaluated the characteristics of each detected cause (there were a total of 2247 causes) in order to evaluate whether the causes of problems detected in the retrospectives of each team remained similar between the treatments. We classified the detected causes by using a classification system developed for analyzing the characteristics of the causes of software project problems introduced in our prior studies (Lehtinen and Mäntylä, 2011; Lehtinen et al., 2014a). The classification system divides the causes based on their types and process areas. In the classification system, a process area (a total of six process area variables) expresses where the cause occurs (see Table 3) whereas a cause type (a total of 14 cause types variables) describes what the cause is (see Table 4). The combination of the process area with the cause type results in *a characteristic of the cause* (a total of $6 \times 14 = 84$ characteristics). For example, if the cause is classified into the *management work* process area and its type is classified as *values & responsibility*, the characteristic of the cause is *values & responsibility in the management work*.

In order to evaluate whether the characteristics of the causes were similar between the treatments, we calculated the correlation between the numbers of causes with the same characteristic over the treatments. The correlation was calculated between the treatments of each team and between all teams combined together. The closer the correlation is to 1, the more similar are the characteristics.

3.4.3. Data from questionnaires

The analyses on the perceptions of participants were partially based on questionnaires. Questionnaire 1 (see Appendix A) was used for both treatments separately. Our aim was to evaluate whether similar parts of the treatments were evaluated similarly. We also evaluated whether different parts of the treatments, i.e. the technique used to organize and visualize the causes, were evaluated differently. Furthermore, after the second retrospective, the participants were asked to compare the treatments by using Questionnaire 2 (see Appendix B). Our aim was to evaluate which treatment the participants prefer the most in the RCA of retrospectives.

Questionnaire 1 included 19 questions covering all phases of the retrospective method. We asked the participants to evaluate the method used to collect the causes of problems. We also asked them to evaluate the method used to organize the causes. Additionally, the questions included statements about the treatments which the participants were supposed to either agree or disagree with. The scale in each question was ordinal and symmetric, e.g., 1 = very bad, 2, 3, 4 = neutral, 5, 6, 7 = very good. We assumed that the evaluations on the treatments vary only in the specific questions about the method used to organize the causes. This was due to the fact that the causes were organized differently, but collected similarly in both treatments (see Section 3.3.1). We compared the treatments by using the Wilcoxon Signed Rank Test with alpha level 0.05 over the evaluations of individual respondents. We also used the Bonferroni correction to calculate the required level of statistical significance. There were a total of 19 questionnaire items. Therefore, the Bonferroni correction gives that the level of statistical significance requires $p = 0.0026$ (0.05/19). The evaluations of participants who were not present at both retrospectives (10 of 61 participants) were excluded from the comparison.

Questionnaire 2 included statements about both retrospectives that the participants were asked to either agree or disagree with. The statements compared the treatments. The scale of the questionnaire was ordinal and symmetric (1 = fully disagree, 2, 3, 4 = neutral, 5, 6, 7 = fully agree). We compared the share of participants who disagreed with the statements to those who agreed with them. The evaluations of participants who were not present at both retrospectives (10 of 61 participants) were excluded from the comparison.

3.4.4. Data from group interviews

In order to consolidate the results from the questionnaires and create a deeper understanding about the perceptions of participants in both treatments, we carried out a group interview with each participating team after the second retrospective. The interview took place immediately after the participants had answered the questionnaires. We did not want to focus the interviews on any specific questions.

Instead, we wanted to create an understanding on what the participants thought about the treatments on a general level. The group interview was open ended (Yin, 1994) and it was started by asking *"which of the used techniques do you prefer the most in the RCA of retrospectives?"* Thereafter, depending on the answers of the participants, the interviewer (the first author) asked clarifying questions about the treatments, e.g., *"why do you prefer the structural list as a more feasible technique?"*

The interviews were transcribed and thereafter coded by the first author. Additionally, the interviews were translated into English. After the interviews were transcribed into a literal form, the interviews were carefully scrutinized. Thereafter, we created categories that conceptualized the comments of the participants. The first author created preliminary categories, which were thereafter reviewed by other authors.

Open coding technique (Flick, 2006) was used to analyze how the participants described the treatments. As suggested in Flick (2006), we started the qualitative analysis by recognizing "the units of meaning", i.e. concepts that reflected the reasoning given in the comments (single words and short sentences of words from the comments). For example, there was a comment *"with CED it is easier to outline the aggregation of causes"*. This comment resulted in a concept: "supports outlining aggregations". Similar concepts were grouped together. Thereafter, all comments were attached to the concepts.

The comments were classified line-by-line to the concepts we recognized, as recommended in Flick (2006). Simultaneously, the comments were divided between the treatments. Thus, we were able to compare how the participants described the treatments on the conceptualized level. In order to compare the comments on a more abstract level, we continued the analysis procedure by recognizing categories that linked the concepts together (Flick, 2006). This was done by pondering the potential meaning of concepts for retrospectives. For example, we assumed that the concepts "supports outlining aggregations" and "supports thinking" would affect the sense making while the participants try to understand the causes of problems in retrospectives. Thus, a category "sense making" was created and the corresponding concepts were linked under it.

The treatments were compared based on the categories and concepts that we recognized. We compared the treatments in order to recognize the concepts that were unique and common for the treatments. This helped us to make comparison and generalize how the treatments were described, which thereafter helped us to make hypotheses about the study results considering the cause count and causal structures, too. Additionally, this helped us in interpreting the evaluation results from the questionnaires. Furthermore, we also compared the number of groups and comments on the related concepts. This was also somewhat important as it indicated the commonality of the perceptions of participants.

4. Results

In this section, we present the study results. We start in Section 4.1 by introducing the quantitative results on the output of the treatments. These include the comparison of the cause count, causal structures, and characteristics of detected causes. Thereafter, in Section 4.2, we introduce how the participants evaluated and described the treatments.

4.1. Output of root cause analysis

In this section, we present the results regarding the output of RCA when applying the two alternative treatments. Table 5 summarizes the retrospectives of each team. It shows that the analyzed (voted) problems of the retrospectives remained mostly similar in each team. Each team analyzed two problems in both sessions. Altogether, the teams had 17 same problems in the second session than in the first session (out of 22 possible) and only one team had both two problems different in the later session. Furthermore, the table shows that most of the projects aimed to develop mobile applications and web-based systems. The other project topics included a tool for Playstation 3, a database system, and an operating system tool. It seems that the variation in the developed systems or their expected quality did not have a clear impact to the voted problems or comparison results. Nine out of the 11 projects aimed to create production quality system.

4.1.1. Cause count

Table 6 presents the descriptive statistics of the number of detected causes divided into the treatments. These include the average (Mean), standard deviation (Std.), minimum (Min), lower quartile (Q1), median (Med), upper quartile (Q3), and maximum (Max). The table views the statistics from the team and individual levels. The team level compares the treatments by using the number of detected causes in each team. Instead, the individual level compares the treatments by using the average number of detected causes per participants in each team. Fig. 5 presents the boxplots for the number of causes at the team level and Fig. 6 presents the boxplots for the average number of causes per participants.

Table 6
Descriptive statistics of the number of detected causes between the treatments.

Level	Treatment	Mean	Std.	Min	Q1	Med	Q3	Max
Average per team	SL	94	22	59	82	92	99	137
	CED	107	22	69	90	111	124	137
Average per participant	SL	17	5	10	15	16	17	27
	CED	20	4	12	17	21	23	26

Table 5
Statistics about the retrospectives.

Team	System	Expected quality	CED						SL					
			#	L	Voted problems	Σp	Σc	c/p	#	L	Voted problems	Σp	Σc	c/p
1	Mobile app	Production	1	F	Co-operation, management	5	76	15	2	F	Co-operation, management	4	70	18
2	Mobile app	Prototype	1	F	Scope, quality	7	87	15	2	F	Quality, scope	6	59	10
3	Web	Production	2	E	Scope, development	5	93	19	1	E	Co-operation, management	6	78	13
4	Web	Production	1	F	Scope, quality	6	127	21	2	F	Quality, scope	5	85	17
5	Playstation tool	Production	1	F	Co-operation, customer	6	137	23	2	F	Quality, customer	6	92	15
6	Web	Production	1	F	Tasks, motivation	5	121	24	2	F	Motivation, skills	5	137	27
7	Web	Prototype	2	F	Scope, task monitoring	5	111	22	1	F	Task monitoring, scope	6	98	16
8	Mobile app	Production	2	E	Process, skills	6	109	18	1	E	Process, skills	6	97	16
9	Database system	Production	2	F	Management, co-operation	5	129	26	1	F	Co-operation, management	5	125	25
10	Operating system tool	Production	1	E	Requirements, risk management	6	69	12	2	E	Requirements, skills	6	90	15
11	Mobile app	Production	2	F	Co-operation, management	5	113	23	1	F	Co-operation, management	6	100	17
					Mean	6	107	20			Mean	6	94	17

#: the first (1) or second (2) retrospective; L: used language (F: Finnish, E: English), Σp: the number of participants, Σc: the number of detected causes, c/p: the number of detected causes per participant.
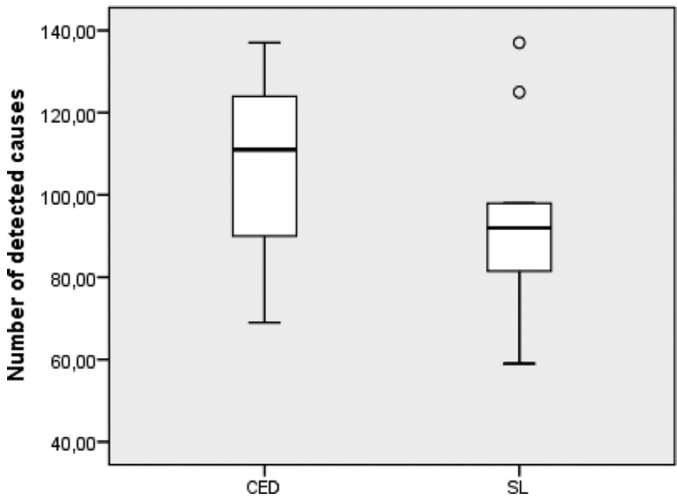
Fig. 5. Boxplot of the number of causes in each team between the treatments.
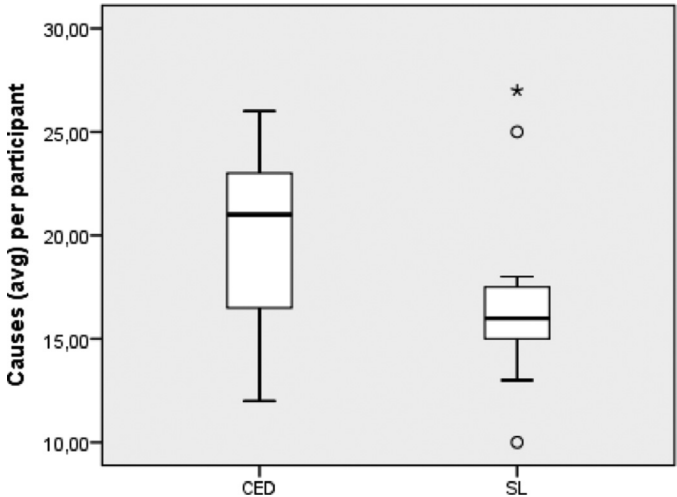

Fig. 6. Boxplot of the number of causes per participant in each team between the treatments.

The descriptive statistics indicate that CED outperformed the structural list (SL) in the cause count (see Table 6, and Figs. 5 and 6). CED resulted in 107 detected causes as an average per team. Respectively, the structural list resulted in 94 detected causes. The mean difference and the 95% confidence interval are 12.8 and ± 13.8, respectively. The effect size between the treatments is medium (Cohen's $d = 0.57$, $p = 0.065$). When analyzing the cause count difference on the team level, CED outperformed the structural list in nine out of the eleven teams (see Table 5 for details).

When we normalize the number of detected causes by the number of participants, we find that in CED the average number of detected causes per participant was 20 compared with 17 in the structural list. The mean difference and the 95% confidence interval are 2.5 and

± 2.69, respectively. The effect size is medium (Cohen's $d = 0.52$, $p = 0.065$). Furthermore, when analyzing the average cause count per number of participants in a team level, CED outperformed the structural list in eight out of the eleven teams (see Table 5 for details).

Thus, whether or not we normalize for the number of participants CED provides a medium effect size in the number of detected causes (Cohen's $d = 0.57$ or $d = 0.52$), but the difference is not statistically significant (alpha $p = 0.05$) due to small sample size ($n = 22$).

4.1.2. Causal structures

Considering the causal structures, Fig. 7 shows the average size of the depth levels (SoDL), see Section 3.3.2. With CED, the SoDL increases between the first and third depth levels. Instead, with the structural list the SoDL increases only between the first and second depth levels. The differences between the treatments in the size of the first ($p = 0.293$, Cohen's $d = -0.51$) and second ($p = 0.811$, Cohen's $d = 0.12$) depth levels are not statistically significant. The effect sizes are medium to small, respectively. Instead, the difference in the size of the depth level three is statistically significant ($p = 0.020$) and the effect size is large (Cohen's $d = 1.01$). Thus, it is possible that CED allows creating causal structures that have more causes starting from the third level than the ones created with the structural list. The difference in the total amount of the detected causes summed from the third to last depth level is medium (Cohen's $d = 0.64$, $p = 0.07$). However, the differences between the treatments in the number of the detected causes at the later depth levels (four to nine) are not statistically significant.

Fig. 8 presents a boxplot of the percentage of hub causes (PoH) in both treatments (a cause that explains more than one effect, see Section 3.3.2). While comparing the proportion of hub causes between the treatments, the t-test gives a large and significant difference ($p = 0.010$, Cohen's $d = 1.42$). As an average, 7.5% (Std. 3.5 percentage points) of the detected causes were hub causes when CED was
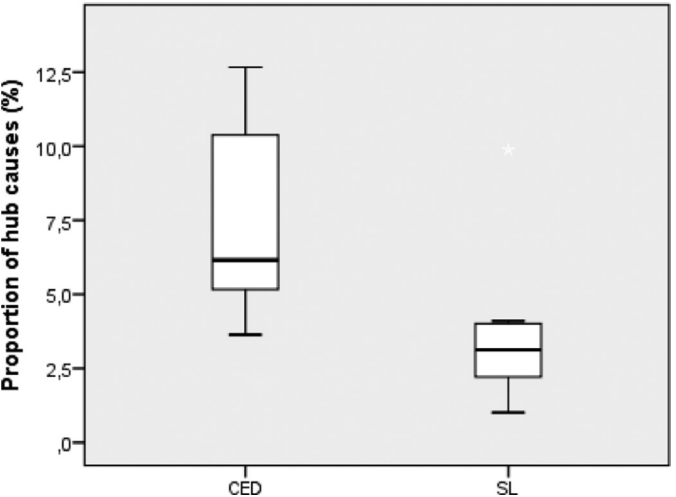
Fig. 8. Boxplot of the proportion (%) of hub causes from all detected causes in the treatments.

Causes (avg) at Depth Levels (1-9)	1	2	3	4	5	6	7	8	9
SL	13	25	23	16	10	5	3	3	
CED	11	25	28	19	15	8	3	1	1
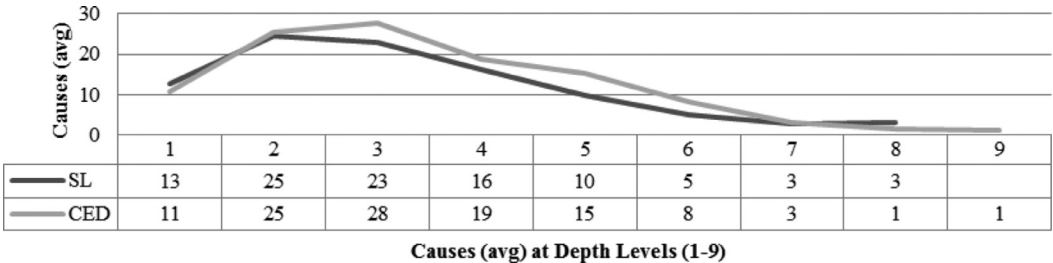
Fig. 7. Summary of the average number of causes (a total of 2247 detected causes) at depth levels (a total of nine depth levels).
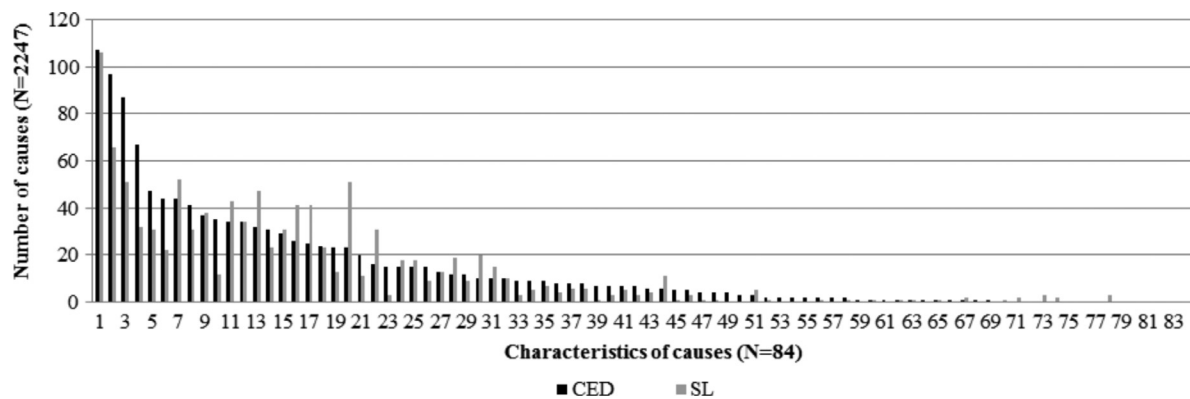
Fig. 9. Distribution of causes among their characteristics.

used, in comparison to only 3.5% (Std. 2.3 percentage points) when the structural list was used.

4.1.3. Characteristics of detected causes

Fig. 9 indicates that similar causes were detected in both treatments. For example, in both treatments the top cause was the output of management work ($n = 106$ for the structural list, $n = 107$ for CED). The figure compares the characteristics of all detected causes (see Section 3.4.2) divided between the treatments. Based on the number of causes with similar characteristics, the data is organized from the highest to the lowest number of characteristics occurred in CED.

Fig. 10 has the same data as Fig. 9 and it illustrates the linear correlation of the number of causes with the same characteristics between the treatments. Each plot in Fig. 10 represents the number of causes with the same characteristic in both treatments. The X-axis shows the number of causes with a certain characteristic of the structural list and the Y-axis shows the number of causes with the same characteristic of CED. The shares of detected causes with similar characteristics correlate strongly between the treatments (Pearson's $r = 0.896, p < 0.001$). This means that the characteristics of the detected causes did not depend significantly on the treatments.

4.2. Feedback of participants

In this section, we present the analysis of the most relevant questionnaire data in terms of the research questions. Next, we
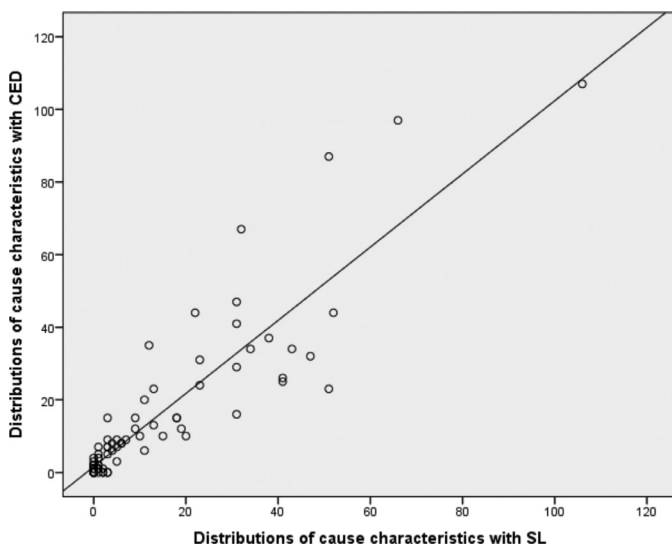

Fig. 10. Linear correlation on the numbers of causes with the same characteristics between the treatments. (A plot in the figure represents the same cause characteristic with both treatments.)

present the participant's evaluations on the methods after each treatment, their comparisons on the two treatments as well as the findings from the group interviews.

4.2.1. Evaluations after each treatment

Table 7 summarizes the results from Questionnaire 1 that had four Topics. This questionnaire was given after both the first and second retrospective. For both treatments, the evaluations were highly similar considering the Topic 1, how the causes of problems were collected. Furthermore, no differences were detected in Topic 3, the general usefulness of the retrospective, or in Topic 4 that measured the social atmosphere of the retrospective.

Topic 2 of the survey evaluated how the detected causes were organized and these questions reflected some differences between the methods. The participants preferred CED when asked about the *technique used to organize the causes* (see Table 7, ID 2.1) and Wilcoxon Signed Rank Test (WSRT) showed that the difference between the treatments is statistically significant ($p = 0.001$). The participants also thought that getting the "big picture of the problem causes" was easier with CED (see Table 7, ID 2.2). However, the difference is not statistically significant (WSRT $p = 0.089$). Finally, the participants saw no difference between treatments in the easiness to register problem causes (see Table 7, ID 2.3) (WSRT $p = 0.464$).

4.2.2. Comparison of the treatments

At the end of the second retrospective, the participants were asked to compare the treatments by using Questionnaire 2, see Table 8. Questionnaire 2 included statements about the retrospectives (first or second "session") which the participants were supposed to agree or disagree on a 7-point ordinal scale from "fully disagree" to "fully agree". We counted the answers of participants being present at both treatments ($N = 51$). The questionnaire asked the participants to evaluate the easiness to register, organize, and outline the detected causes. The questionnaire also asked to agree or disagree whether or not RCA should be conducted by using CED instead of using the structural list. Table 8 summarizes the answers of the participants divided into those who used CED and those who used the structural list (SL) in the second retrospective session. It seems that the retrospectives using CED were perceived as easier regarding *registering*, *organizing*, and *outlining* the detected causes. Additionally, most of the participants perceived that RCA should rather be conducted with CED than the structural list (a total of 75%). It is possible that this result is biased toward CED due to the somewhat loaded statement in Questionnaire 2.

4.2.3. Results from the group interviews

Table 9 summarizes the arguments that were acquired from the group interviews to describe the treatments. The concepts that we

Table 7
Summary of feedback from Questionnaire 1 (**bold** indicates the preferred technique).

Topic	Technique	N	Answers on scale (%)[a]							Median
			1	2	3	4	5	6	7	
1. Collecting the causes (no variation in the method)										
1.1 Difficulty to detect problem causes	CED	51	–	7.8	15.7	9.8	29.4	33.3	3.9	5
	SL	50	–	10.0	10.0	16.0	36.0	22.0	6.0	5
1.2 Easiness to collect causes	CED	51	–	–	3.9	13.7	21.6	43.1	17.6	6
	SL	50	–	–	2.0	10.0	32.0	38.0	18.0	6
1.3 The method used to collect causes	CED	51	–	–	–	–	9.8	64.7	25.5	6
	SL	51	–	–	–	2.0	9.8	56.9	31.4	6
1.4 Usefulness of cause collection	CED	51	–	–	–	2.0	17.6	43.1	37.3	6
	SL	48	–	–	–	4.2	8.3	54.2	33.3	6
1.5 Importance of collecting sub causes	CED	51	–	–	–	5.9	17.6	45.1	31.4	6
	SL	50	–	–	–	2.0	16.0	62.0	20.0	6
2. Organizing the causes (variation in the method)										
2.1 The method used to organize causes	**CED**	51	–	–	2.0	2.0	19.6	58.8	17.6	6
	SL	51	–	–	2.0	9.8	45.1	31.4	11.8	5
2.2 Difficulty to get the big picture of problem causes	**CED**	51	5.9	17.6	27.5	17.6	19.6	9.8	2.0	3
	SL	50	6.0	14.0	18.0	12.0	30.0	14.0	6.0	4.5
2.3 Easiness to register problem causes	CED	49	–	4.1	8.2	22.4	38.8	22.4	4.1	5
	SL	50	2.0	6.0	12.0	20.0	20.0	32.0	8.0	5
3. Retrospective in general										
3.1 Cost-efficiency of the workshop	CED	50	–	–	2.0	4.0	24.0	36.0	34.0	6
	SL	49	–	–	4.1	4.1	30.6	34.7	26.5	6
3.2 Eff. in comparison to other methods	CED	39	–	–	–	10.3	12.8	38.5	38.5	6
	SL	38	–	–	–	5.3	18.4	36.8	39.5	6
3.3 Usefulness for corrective actions	CED	49	–	–	–	4.1	28.6	42.9	24.5	6
	SL	48	–	–	4.2	2.1	12.5	62.5	18.8	6
3.4 Usefulness of workshop in general	CED	51	–	–	–	2.0	9.8	39.2	49.0	6
	SL	48	–	–	2.1	4.2	12.5	41.7	39.6	6
3.5 This workshop was waste of time	CED	51	56.9	31.4	5.9	3.9	2.0	–	–	1
	SL	48	52.1	31.3	10.4	6.3	–	–	–	1
3.6 Correctness of detected causes	CED	51	–	–	–	8.0	6.0	58.0	28.0	6
	SL	51	–	–	–	3.9	29.4	35.3	31.4	6
3.7 Solvability of detected causes	CED	48	–	–	2.1	25.0	37.5	29.2	6.2	5
	SL	50	–	–	4.0	24.0	28.0	38.0	6.0	5
4. Social atmosphere (team dependent)										
4.1 Communication openness	CED	51	–	–	–	–	7.8	31.4	60.8	7
	SL	48	–	–	–	2.1	4.2	37.5	56.3	7
4.2 My team's effort	CED	50	–	–	–	–	2.0	56.0	42.0	6
	SL	49	–	–	–	2.0	10.2	57.1	30.6	6
4.3 My personal effort	CED	48	–	–	2.1	4.2	41.7	43.8	8.3	6
	SL	51	–	–	2.0	19.6	25.5	49.0	3.9	6
4.4 Team spirit	CED	51	–	–	–	5.9	23.5	35.3	35.3	6
	SL	48	–	4.2	–	8.3	4.2	52.1	31.3	6
4.5 Team members purposefully hided causes	CED	50	28.0	34.0	14.0	16.0	8.0	–	–	2
	SL	47	27.7	29.8	19.1	14.9	4.3	2.1	2.1	2
4.6 Team members did not dare to present all causes	CED	49	22.4	36.7	10.2	18.4	8.2	4.1	–	2
	SL	49	18.4	34.7	22.4	6.1	14.3	4.1	–	2

CED: Cause–effect diagram; SL: structural list; N: number of respondents.
[a] The scale was: 1=very low; 2, 3, 4=neutral, 5, 6, 7=very high.

Table 8
Comparison of the treatments from Questionnaire 2 (**bold** indicates the preferred technique).

Statement	Second session	N	Answers on scale (%)[a]							Median
			1	2	3	4	5	6	7	
1. Registering the causes was easier in the first session	**CED**	21	4.8	28.6	23.8	23.8	4.8	14.3	–	3
	SL	30	3.3	13.3	13.3	16.7	16.7	20.0	16.7	5
2. Registering the causes was easier in this second session	**CED**	21	–	–	–	9.5	28.6	52.4	9.5	6
	SL	30	6.7	13.3	30.0	20.0	10.0	20.0	–	3.5
3. Organizing the causes was easier in the first session	**CED**	21	14.3	47.6	23.8	14.3	–	–	–	2
	SL	29	6.9	13.8	13.8	3.4	27.6	10.3	24.1	5
4. Organizing the causes was more difficult in this second session	**CED**	21	14.3	42.9	23.8	14.3	–	–	4.8	2
	SL	29	6.9	24.1	10.3	13.8	27.6	6.9	10.3	4
5. The number of causes created difficulties in the first session	**CED**	21	–	9.5	14.3	9.5	42.9	14.3	9.5	5
	SL	30	3.3	16.7	16.7	13.3	23.3	20.0	6.7	4.5
6. The number of causes created difficulties in the second session	**CED**	21	4.8	33.3	28.6	19.0	9.5	4.8	–	3
	SL	30	6.7	16.7	16.7	16.7	33.3	6.7	3.3	4
7. Outlining the causes was easier in this second session	**CED**	21	–	–	–	28.6	23.8	38.1	9.5	5
	SL	30	13.3	10.0	30.0	13.3	20.0	13.3	–	3
8. RCA should rather be conducted by using CED	**CED**	21	–	4.8	–	4.8	4.8	33.3	52.4	7
	SL	30	–	6.7	20.0	10.0	16.7	10.0	36.7	5

[a] The scale was: 1=fully disagree; 2=disagree, 3=somewhat disagree, 4=neutral; 5=somewhat agree, 6=agree, 7=fully agree.

Table 9
Comparison of the arguments used for describing the cause and effect structuring techniques.

Category	Concept	CED	SL
Sense making	Supports outlining aggregation	*With CED it is easier to outline the aggregation of causes:* the number of comments (8) and groups (6).	*With the list it is easier to interpret the causes if the causes are not much interconnected:* the number of comments (1) and groups (1).
	Supports outlining causal relationships	*With CED it is easier to outline the causal relationships:* the number of comments (15) and groups (8).	–
	Supports thinking	*There is no list of causes in my brains, instead, there are causal relationships:* the number of comments (3) and groups (3).	*I consider these causes as a top-down list in my brains and thus the list is more feasible for me:* the number of comments (1) and groups (1).
	Supports discussion	*I think that CED improved discussion in the session:* the number of comments (2) and groups (1).	*While registering the causes less time is used to formalism, which improves the discussion:* the number of comments (2) and groups (1).
Ease-of-use	Easier to use in general	*CED is easier to operate:* the number of comments (5) and groups (3).	*I experienced the list approach more lightweight than CED:* the number of comments (9) and groups (5).
	Easier to read	*CED is much easier to read than the list of causes:* the number of comments (2) and groups (1).	*The list approach results to more readable structure:* the number of comments (8) and groups (6).
	Easier to find registered causes	*It was relatively easy to find the causes already detected from CED whereas it was difficult from the list structure:* the number of comments (3) and groups (2).	*The list structure can visualize higher number of causes simultaneously helping to find causes already detected:* the number of comments (1) and groups (1).
	Easier to organize	*I think that less time is used to organize the causes with CED:* the number of comments (1) and groups (1).	*I assume that less time is used to organize the causes with the list:* the number of comments (1) and groups (1).
	Easier visual structure	*The structure of CED is much more feasible:* the number of comments (16) and groups (7).	–
	Easier to navigate	*CED is easier to navigate:* the number of comments (4) and groups (4).	–
Accuracy	Increases efficiency	*I assume that the graph structure helps to detect causes more efficiently:* the number of comments (6) and groups (4).	*The list approach requires less time while the causes are organized, which makes it more efficient:* the number of comments (2) and groups (2).
	Increases accuracy	*I think that with CED it is easier to focus on specific branches:* the number of comments (3) and groups (2).	–
	Increases systematics	*It was easier to contribute to CED as I was able to process the causes detected more systematically:* the number of comments (2) and groups (2).	–

recognized indicated different pros and cons between the treatments. While the participants perceived that CED outperforms the structural list in its visual structure, they also perceived that the structural list (SL) outperforms CED in its readability.

From the interviews, we recognized three high level categories that linked the comments of participants together. These included *Sense making*, *Ease-of-Use*, and *Accuracy*. Sense making is about comments that describe how the treatments helped the participants to understand how the detected causes affect the problem together. Ease-of-Use is about comments that describe how the treatments helped the participants to use the cause and effect structuring technique. Accuracy includes comments that describe how the treatments helped the participants to detect causes.

The participants perceived that CED outperforms the structural list in Sense making and Accuracy. It was perceived that CED supports outlining the aggregations of causes (6 groups) and causal relationships (8 groups). Furthermore, the visual structure of CED was perceived as feasible for RCA (7 groups) and especially an easier technique to navigate the detected causes (4 groups). Additionally, the participants perceived that CED helped focusing on specific causes (2 groups) and it was easier to process the detected causes systematically (2 groups).

The participants also found the structural list as useful. It was reported that the structural list makes it easier to read the detected causes (6 groups). It was also claimed that the high readability makes the structural list lightweight and thus it increases the efficiency of the analysis (2 groups). However, CED was perceived as increasing efficiency more often (4 groups). The participants also claimed that the structural list is generally easier to use (5 groups). On the other hand, many participants reported the opposite (3 groups).

5. Discussion

In this section, we answer the research questions, compare our findings with prior works and outline possible threats to the validity.

5.1. RQ1: Is there a difference between the techniques in terms of the outcome of RCA?

This research question was studied with three sub-questions. Below we summarize the answers. *RQ1a: Is there a difference in the number of the detected causes?* Our results in Section 4.1.1 showed that in nine teams out of 11 CED found more causes (avg. 107) than the structural list (avg. 94) and the difference between the treatments has medium effect size ($d = 0.57$). Thus, the teams performed more active knowledge sharing with CED. However, the difference is not statistically significant due to small sample size. Thus, we interpret that our results give only weak evidence in favor of using CED in retrospectives. The participants evaluated that the detected causes were equally "correct" and "solvable" in both treatments (see Table 7). Respectively, both treatments resulted in active retrospective meetings, where the participants eagerly presented and shared their visions about the software project, which is important for retrospectives (Dingsøyr, 2005). Therefore, we conclude that the observed small increase in the amount of detected causes favors the use of CED, but does not alone warrant a strong recommendation for using CED over the structural list in project retrospectives.

RQ1b: *Is there a difference in the structures of the detected causes?* Our results in Section 4.1.2 showed that the number of causes increased between the first and third depth levels when using CED. Instead, for the structural list, the number of causes increased only among the first and second depth levels. The difference in the size of

the third depth level is large and statistically significant. Therefore, we hypothesize that CED allows creating cause–effect networks that have more detected causes starting from the third level than ones created with structural list (a total of 75 vs. 60 detected causes on average), see Fig. 7. Our interpretation of this is that CED encourages toward the deeper investigation of causes than the structural list, and thus, using CED can be beneficial if understanding the cause–effect structure of the problem requires deeper analysis than one or two levels of causes.

The use of CED also increased the proportion of hub causes. As an average, 7.5% of the causes detected with CED explained more than one effect, whereas the proportion of such causes was only 3.5% when the structural list was used. The difference between the treatments is statistically significant and large. This suggests that CED enables the participants to link causes to each other more effectively. Thus, the knowledge created by CED is richer compared with the structural list that creates a more fragmented view for the participants. This finding indicates that CED helps to create more comprehensive understanding on the underlying problems, which is important for making inferences and self-explanation efficiency, as discussed in Section 2.2. The finding consolidates the experimentation results of Ainsworth and Th Loizou (2003) who presented that the use of diagrams encourages individuals to create "mental images" on the cause and effect relationships, which helps them to explain the studied system of knowledge as a whole, increasing the efficiency of learning.

RQ1c: Is there a difference in the characteristics of the detected causes? Our results in Section 4.1.3 showed that the treatments did not have a high impact on the characteristics of the causes, e.g., with both approaches the top cause was characterized as the output of management work. The shares of detected causes with similar characteristics correlated strongly between the treatments. This result means that the techniques used to organize and visualize the causes have no effect on the characteristics of the detected causes. Thus, the effect of these techniques for learning about the occurrence of different types of problems remains somewhat similar.

A generally interesting perennial question of RCA is the impact it has on the practice. Our results show that similar voted problems were analyzed and similar cause characteristics appeared in the first and second retrospective session. The similarities in the problems and their cause characteristics may be viewed as lack of impact on the part of the method, because the participants are analyzing similar problems and detecting similar causes in both sessions. The similarity of cause characteristics was high in the full data set (correlation $r = 0.896$) indicating no difference between the sessions. However, individual team level correlation was lower ($r = 0.575$), which suggest higher variance at a team level. In addition, the data from Questionnaire 1 shows that the correctness and solvability of the detected causes were perceived high (Table 7 rows 3.6 and 3.7). These data sets suggest that RCA had impact on the team level. On the other hand, we acknowledge that fully solving the complex problems in the few weeks the teams had between the sessions is very challenging. Our plan is to research the impact of RCA in longitudinal industrial studies.

In summary, the only significant difference between the techniques, regarding the RCA outcome, seems to be that CED increases the number of presented interconnections between the detected problems of software projects. Our research in industrial context has identified such ability as very important for understanding the causes of software project failures (Lehtinen et al., 2014a), which represent complex software engineering problems that cannot be solved by considering the shallow causes only.

5.2. RQ2: Do the perceptions of retrospective participants vary between the techniques?

This research question was studied with two sub-questions. RQ2a: *Is there a difference in the preferred technique?* The results from Ques-

tionnaire 1 indicate that the retrospective utilizing CED was perceived generally as a better technique to organize the detected causes. CED was evaluated as a "good" technique to organize the detected causes whereas the structural list was evaluated as "somewhat good" (see Section 4.2.1). Similarly, the results from Questionnaire 2 indicate that the participants preferred using CED in the RCA of retrospectives. Furthermore, our results indicate that outlining the detected causes is easier with CED. Despite the difference between the treatments was not statistically significant ($p = 0.089$), it was consolidated in the interviews and Questionnaire 2. In Questionnaire 2, CED was perceived as easier regarding registering, organizing, and outlining the detected causes. In the interviews, most of the teams reported that CED made it easier to outline the detected causes. These results indicate that using CED in the RCA of retrospectives is reasonable as the retrospective participants prefer using it. However, also the structural list helps to organize the causes of problems. Additionally, it is not perceived significantly different than CED when the participants evaluate the outcome of RCA. Furthermore, the techniques did not make any difference to the perceptions on the retrospective meetings in general. For both techniques, the meetings were perceived equally cost-efficient and useful for corrective action innovation.

RQ2b: How do the retrospective participants evaluate and describe the techniques? Considering the similarities between the treatments, the results from the group interviews (see Table 9) indicated that the participants perceived both treatments as feasible for registering the causes. The results from Questionnaire 1 consolidate this assumption. The participants agreed for both treatments similarly that it was easy to register the detected causes among the other causes. It is possible that this similarity was due to the fact that the facilitator was the one who registered the detected causes among the other causes based on the instructions of the participants (see Section 3.3.3).

Considering the differences between the treatments, the participants emphasized that CED outperforms the structural list when the detected causes are outlined. The visual structure of CED was described as "feasible for RCA". It helped outline the aggregations of causes and made it easier to outline the perceived cause and effect relationships, which could also explain why CED resulted into increasing proportion of hub causes. The participants claimed that CED was easy to navigate and operate. Thus, it was also easier to focus on the detected causes. Therefore, the participants perceived that CED increases the accuracy of the analysis and it improves sense making of the detected causes. Similar claims have been presented in the prior studies. For example, Larkin and Simon (1987) discussed about the location of information in a diagrammatic representation and claimed that in diagrams the needed information is "present and explicit at a single location", which helps the learner to search, recognize and make inference about the studied system on knowledge.

There were arguments that support using the structural list, too. The participants claimed that the visual structure of the structural list allows more causes to be visible at the same time. The structural list was also described as easier to operate due to its high readability, as indicated by Ottensooser et al. (2012). Interestingly, it was claimed that the visual structure of the structural list is beneficial only if the number of detected causes remains low. Similar conclusion can be made based on the quantitative analysis of the size of depth levels (see Section 5.1). Moody (2009) stated that "different representations of information are suitable for different tasks and audiences". Based on prior studies (McLeod and MacDonell, 2011), software project problems are complex and they are often related to many causes. Respectively, the positive effect of CED for learning has been determined especially with complex problems (Ainsworth and Th Loizou, 2003). Thus, we hypothesize that the use of CED becomes increasingly beneficial when the complexity of analysis increases.

To conclude, there seems to be a difference between the techniques considering the perceptions of retrospective participants. In terms of organizing a high number of problem causes, the

participants perceived that CED provided more flexible and visually attractive structure. Similar conclusion has been given by Bjørnson et al. (2009). Additionally, when making sense about the causes of problems, the participants perceived that CED helped to navigate the detected causes. Such ability has been related to CED also in a prior study (Larkin and Simon, 1987). We assume these success factors of CED explain why the participants also experienced that the use of CED provided additional value for their software project retrospectives. Combining this conclusion with the actual outcome of the retrospectives indicates that CED is a better technique for RCA than the structural list. Despite that it does not really matter if one method allows people to identify slightly more causes than the other, it could be more important in practice if the participants perceive the method as better and more attractive. Our results indicate that CED could bring additional value to the retrospective meeting and increase the motivation of the team members to conduct one.

5.3. Comparison to prior works

Lee et al. (1992) claimed that sharing cognitive maps, which include perceived cause and effect relationships between actions and their responses, results in organizational learning. The maps that they introduced follow the visual structure of CED. Our results support the recommendations of Lee et al. CED could outperform the structural list technique when the team is trying to learn from their problems. Our results indicate that the use of CED helps in creating linkages between the causes of problems, which has been claimed to be the key for self-explanation efficiency (Ainsworth and Th Loizou, 2003). This finding indicates that the use of CED brings additional value to the retrospectives, which consolidates the prior studies recommending using CEDs in the RCA of retrospectives (Anbari et al., 2008; Bjørnson et al., 2009; Dingsøyr, 2005; Lehtinen et al., 2011). However, we acknowledge that the amount of "learning" is very hard to measure, especially, with the techniques directly in connection to the retrospective meeting including the cause count, the size of depth levels, and the proportion of hub causes. Thus, our results regarding the "amount of learning" are limited.

Recently, Bjarnason et al. (2014) presented a timeline approach to conduct retrospectives. They propose an evidence-based timeline to fuel discussions and share experiences in the retrospective session. The timeline is also an example of a graphical approach used in retrospectives. The timeline itself represents potential cause–effect relationships through a temporal sequence of events, even though the cause–effect relationships are not explicitly created. Thus, merging the traditional CED approaches with evidence-based timelines could provide even a more accurate picture of the events and enable better learning in the reflection meetings. The external representation could also improve the post-retrospective activities. In comparison with textual representation, diagram representation could be easier to remember (Ainsworth and Th Loizou, 2003; Larkin and Simon, 1987) and therefore it becomes more optimal for knowledge sharing.

Considering alternative techniques to create CED (Burnstein, 2003; Stevenson, 2005; Andersen and Fagerhaug, 2006; Ishikawa, 1990; Bjørnson et al., 2009; Nakashima et al., 1999; Latino and Latino, 2006; Ammerman, 1998; Andersen and Fagerhaug, 2006; Rooney and Vanden Heuvel, 2004), it seems evitable that in software project retrospectives the diagramming technique should support network structures (Lehtinen et al., 2011). This is because of the hub causes (Bjørnson et al., 2009) (in our study their proportion was 7.5% as an average). Duplicating the same cause many times decreases the comprehensibility of the external representation having a negative impact to Search and Recognition (see Section 2.2). The fishbone diagram includes the same problem, as it is a tree structure (Lehtinen et al., 2011).

Bjørnsson et al. (2009) compared two CED techniques with a controlled student experiment and showed that using the fishbone dia-

gram in RCA resulted in lower number of detected causes when compared with the directed graph. We had a similar finding about the structural list, but the difference in the number of detected causes was not as large as was reported by Björnsson et al. (2009). One explanation for this difference could be the RCA facilitator of the retrospectives. Björnsson et al. (2009) assumed that the difference might have been smaller if they had used professional facilitators. Another explanation could be the method used to collect and register the causes. The method that we used did not change between the treatments, whereas the prior experiment used *"a nominal brainstorming technique"* with the directed graph and *"an interactive technique"* with the fishbone diagram (Bjørnson et al., 2009). Furthermore, in contrast to the structural list technique, the fishbone diagram steers the participants to classify the detected causes during the analysis (Lehtinen et al., 2011). Such a categorization is also known as "modularization" (Moody, 2009), used to manage the complexity of raw data. It is possible that the cause classification decreases the number of detected causes. If the participants are forced to consider the cause classes simultaneously while trying to detect new causes, less new causes are detected because they need to focus on two things simultaneously. On the other hand, modularization likely becomes highly important if the retrospective findings are communicated for other people (e.g., Lehtinen et al., 2014a).

To summarize, it seems that a network structured CED is needed in the RCA of software project retrospectives, because it helps the retrospective participants in explaining and making sense about the perceived relationships of the causes of problems. CED is visually more attractive and technically more effective than the structural list. Additionally, the retrospective participants prefer using CED. These hypotheses are in line with the prior studies which have recommended using CEDs in the RCA of software project retrospectives (Anbari et al., 2008; Bjørnson et al., 2009; Dingsøyr, 2005; Lehtinen et al., 2011). Our hypotheses are also in line with the prior study about the cognitive maps (Lee et al., 1992). Finally, the prior studies indicate that the usefulness of CED is not limited to retrospective meetings only, but to post-retrospective activities where the retrospective findings are shared for other teams and organization members. The diagram representation is a better way to share the findings, because it is easier to learn, it is easier to remember, and it increases the efficiency of self-explanation and inference.

5.4. Evaluation of the research

This section discusses the validity of our results using a validation scheme presented by Runeson and Höst (2008). We will present the construct validity in Section 5.4.1, the internal validity in Section 5.4.2, the external validity in Section 5.4.3, and the reliability of the study in Section 5.4.4.

5.4.1. Construct validity

Construct validity reflects the extent to which the studied operational measures really represent what is investigated according to the research questions (Runeson and Höst, 2008). In this study, the operational measures included the outcome of RCA, questionnaires, and interviews.

In order to analyze the characteristics of detected causes, we used a classification system (see Section 3.4.2). Classifying the causes likely dissipated their dissimilarities and simultaneously highlighted their similarities. This means that there is a risk for the construct validity that the detected causes were not as similar as our results indicated (see Section 4.1.3). Previously, we have qualitatively analyzed the causes which were detected in this study (Vanhanen and Lehtinen, 2014) and we did not note any differences in the detected causes between the treatments. Additionally, during this study, we did not note any differences in the detected causes while using the classification system. Furthermore, there are no good reasons to assume that

the detected causes are significantly different when they are detected with CED versus the structural list.

Considering the evaluations of participants, there is a risk for construct validity regarding the questionnaires. It is possible that the participants understood the questions in the forms differently, and thus their evaluations varied. The items in Questionnaire 2 were somewhat loaded and unclear. It is also possible that some participants were more or less critical than others while making the evaluations. Furthermore, it is possible that the participants did not evaluate the treatments objectively. A total of 61 participants filled in the questionnaires. Additionally, 84% of the participants were present at both retrospectives. We believe that there were enough participants to make a statistical comparison between their evaluations. Table 7 summarized the feedback from Questionnaire 1. The standard deviation between the evaluations was small. Additionally, the participants evaluated similar parts of the treatments similarly and different parts somewhat differently. Thus, it is likely that the participants understood the questions at least somewhat similarly and most of them were objective. Additionally, this means that the questionnaire worked as planned. Furthermore, we used the Wilcoxon Signed Rank Test with alpha level 0.05 to detect systematic differences in the evaluations of an individual respondent. The alpha level was also corrected by using the Bonferroni correction resulting in a required level of statistical significance ($p = 0.0026$). Thus, even if the participants were more or less critical while making the evaluations, we were able to recognize the preferred treatment.

Considering the arguments used to describe the treatments, there is a risk for construct validity regarding the group interviews. It happened that some team members did not state any comments as the other team members dominated the interview. Thus, it is possible that the results from interviews are skewed to the opinions of dominating participants. However, most of the participants from each team provided comments about the treatments. Thus, in order to draw out conclusions and make hypotheses about the treatments, we believe that our results represent the perceptions of participants inclusively enough.

Furthermore, the first author transcribed the interviews and used open-coding to draw out the conclusions. Thus, there is a risk for construct validity regarding the possible misinterpretations of the interviews. However, the qualitative research method that was used (see Section 3.4.4) utilizes the comments and keywords the retrospective participants used while they did the comparison between the treatments. Thus, the conclusions made by the first author are based on the comparisons the retrospective participants made. Additionally, the interviews were conducted for each group separately. Thus, the conclusions are based on many data sources instead of few. The interviews were also video recorded. Thus, while transcribing the interviews, the first author was able to recall the social atmosphere and specific comments about the treatments.

5.4.2. Internal validity

Internal validity is of concern when the causal relations of the measured factors are examined (Runeson and Höst, 2008). In this study, the examination covered the causal relationships between the treatments and response variables.

The research settings of each team were similar in both retrospectives because we controlled the roles of participants, language, physical conditions, the retrospective facilitator, the education background, cultural differences, skills, and differences in ages and sex. We can see from Table 7 that the retrospective participants evaluated the openness in communication, personal effort, team effort, and team spirit similarly in both treatments. They also evaluated that their team members did not significantly hide causes during the retrospectives and they dare to present the detected causes for other team members. Thus, we assume that also the motivation and team spirit remained similar between the treatments. We also controlled

the retrospective method. It was conducted similarly in all retrospectives and the similar parts of the method were also evaluated similarly (see Table 7). The only significant difference in the evaluations was related to the variation in the treatments.

Considering the comparison of the number of detected causes and causal structures, there is a risk for internal validity regarding the specific focus of each retrospective. The specific focus of the retrospectives varied (see Table 5), because the team members voted slightly different problems to be further analyzed with RCA (see Table 5). Thus, there is a risk for internal validity regarding our comparison results on the number of detected causes and causal structures. Considering this risk, most of the teams (seven out of eleven) had a highly similar focus in both of their retrospectives as the voted problems were similar in both retrospectives. Thus, the risk was low in most of the teams. Furthermore, the results from these teams are in line with the results of all teams together. Additionally, the characteristics of the detected causes remained similar in each team (see Section 4.1.3). Thus, even though the voted problems slightly varied, similar causes were recognized in the retrospectives. Therefore, we believe that the voted problems did not make a major bias to the comparison results.

There is a risk for internal validity regarding the number of retrospective participants (see Table 5). In six teams, the number of participants varied +/-1 between the retrospectives. Thus, it was possible that the variation in the number of participants biased the comparison results. We evaluated this risk by calculating the correlation between the number of participants and the number of detected causes. The null hypothesis was that the number of participants in the teams does not correlate with the number of detected causes. We tested both treatments (A and B) separately and together (AB). None of these tests resulted in a significant correlation (Pearson's $pA = 0.658$, $pB = 0.727$, $pAB = 0.566$) and the coefficient values were very low ($rA = -0.151$, $rB = -0.119$, $rAB = -0.129$). Thus, the tests did not reject the null hypothesis. Additionally, the difference between the numbers of participants in treatments was not statistically significant over the teams (WSRT gives $p = 1.000$). Thus, the potential bias in our comparison results caused by the varying number of participants cannot be concluded with these tests.

Furthermore, our results were neither highly dependent on the order of the treatments. For the project teams which started with the structural list, the average number of detected causes was 100 in the first retrospective. When those teams used CED in their second retrospective, the average number was 111, 11% increase as an average. For the project teams which started with CED, the average number of causes was 103. Instead, when those teams used the structural list in the second retrospective, the average number was 89, 14% decrease as an average. Additionally, the project teams which detected a high number of causes with structural list also did that with CED and vice versa. Pearson's correlation between the treatments of each team based on the number of causes is strong ($r = 0.580$, $p = 0.061$) but it is not statistically significant due to the low number of teams ($N = 11$). Furthermore, the correlation between the treatments of each team on the average number of causes per participants is strong and it is also statistically significant ($r = 0.648$, $p = 0.031$). Furthermore, as the change in the number of causes between the treatments was very similar in each team, we conclude that the order of treatments did not violate the comparison results. This also indicates that the risk of learning effect bias in the comparison results is low.

5.4.3. 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 (Runeson and Höst, 2008). Considering the cause count, causal structures, and the perceptions of participants, our results indicate that CED outperforms the structural list in the RCA of retrospectives which are conducted in small software project teams with a skilled

facilitator. We believe that the external validity of this conclusion is high. However, our results are based on the retrospectives of student teams. Thus, there is a risk for external validity regarding the retrospectives which are conducted in industrial software teams. Our results cannot be used to present the absolute level of improvements, but we believe they are valid for representing the improvement trend over the treatments (Runeson, 2003). Our results are also limited to retrospectives where only negative project experiences are analyzed, whereas the prior study considered also positive experiences (Bjørnson et al., 2009). Furthermore, our results are limited to RCA which is conducted by using a monitor and software tool. Thus, we cannot generalize our findings to RCA which is conducted by using a whiteboard and Post-it notes.

In industrial software teams, the number of causes could easily be over a hundred (Lehtinen et al., 2011). Our results indicate that CED improves the effectiveness of retrospectives when a high number of causes are detected. We conducted somewhat similar retrospectives to CED in four software companies covering the work of over 100 employees in each company (Lehtinen et al., 2011). As a result, the lowest number of detected causes was 163, which is significantly more than the number of detected causes in the project teams of this study (see Table 5). Thus, we believe that using CED in these four companies was a more optimal choice than the structural list. Respectively, our recent study with industrial software teams has consolidated this assumption by indicating that the motivation of the teams to conduct retrospectives increase while CED is used instead of writing down structural lists about the problems and their causes (Lehtinen et al., 2014b).

Furthermore, despite our conclusions are based on the retrospectives of small software teams, we believe that our results are also valid in large software teams. We assume that the complexity and cross-functionality of the problems of larger software project teams would increase the number of detected causes. If few causes of the problem are detected, then it is likely that the visualization technique does not make much difference to the retrospective outcome. However, when a high number of causes are detected, then the need to use CED increases.

Considering the perceptions of retrospective participants, we believe that the external validity of our results is also high. A similar conclusion about the RCA method which utilizes CED has been presented (Lehtinen et al., 2011, 2014b; Bjørnson et al., 2009). It has also been claimed that the flexible structure of CED is one of its advantages (Bjørnson et al., 2009). Additionally, our results are not limited to perceptions of a few individual. Instead, our results cover the opinions of dozens of people.

5.4.4. Reliability

Reliability is concerned with the extent to which the data and analysis are dependent on a specific researcher (Runeson and Höst, 2008). Our results are based on quantitative and qualitative data. Considering the quantitative data, there is a risk for reliability as the first author steered the retrospectives. Even though he tried to act as objectively as possible, it is possible that he unconsciously biased the results somehow. We tried to minimize such bias. Each retrospective strictly followed the retrospective method introduced in Section 3.3.1. Respectively, the first author is familiar with RCA and the software tools used in the treatments and thus he did not need to use time to learn to use them properly. We assume that using the same facilitator in each retrospective was an advantage as now the retrospectives are more comparable than they would have been if the facilitators would have changed over the teams or treatments.

Furthermore, there is a risk for reliability regarding the evaluations of participants. It is possible that the personal characteristics of the facilitator affected the evaluations. To control this problem we used the paired design and randomized the starting order of treatments for each team. Additionally, the participants did not know our research

goals in advance, and similar questions were asked in questionnaires after both treatments. Therefore, we were able to analyze how the answers of individual respondents varied over the treatments. Additionally, we underlined for the participants that they should evaluate the treatments as objectively as possible. Furthermore, we used the group interviews to consolidate the results from questionnaires. The results from both data sources are in line with one another.

6. Conclusions and future work

CED is a commonly recommended technique for RCA, as indicated in our earlier literature review (Lehtinen et al., 2011). However, there are no studies where the effectiveness of using CED is compared with the effectiveness of RCA without it. In this paper, we performed a controlled experiment comparing CED with the structural list in the context of project teams ($n = 22$) of a software engineering capstone course. We evaluated the outcome of RCA in software project retrospectives and the perceptions of retrospective participants using CED in comparison to those using the structural list technique. We made three main findings in this research.

First, we found weak evidence that the measured output of CED is better in comparison to the structural list. CED increased the cause count with medium effect size, however, the difference is not statistically significant due to small sample size. The difference was caused by the fact that CED had more causes on the deeper levels than structural lists. Thus, using CED can be beneficial if a problem cannot be solved only by looking at the shallow causes. In addition, the causal structures which were created with CED had higher proportion of hub causes indicating that CED allows the creation of richer understanding about the interconnections between the causes of the problem. This difference was statistically significant with large effect size.

Second, in terms of the perceptions of the retrospective participants, there are significant differences between the techniques. CED was perceived as a better technique in the questionnaires and most of the participants (75%) prefer using CED, instead of the structural list.

Third, the qualitative analysis of both methods showed that both methods had advantages. CED was perceived as a better technique to organize the causes of problems, because it provides a more flexible and visually attractive structure and it is also perceived as easier to navigate when making sense about the causes of the problems. The structural list was seen as easier to read and it could present more causes simultaneously on screen than CED.

Our implications for practice are as follows.

- CED was preferred by the participants. Using CED can increase the motivation to conduct RCA in the project retrospectives.
- CED provides richer analysis on the interrelations of causes and thus, it is preferable in particular for the more complex problems.
- The differences between these techniques are not large, which means the found benefits do not justify enforcing CED on a reluctant project team.
- Drawing a CED requires a specific software tool, in practice, whereas a structural list can be used with a standard text editor.

Obviously, software companies rarely have time to conduct retrospectives (Glass, 2002). However, they are likely valuable and therefore they should also be as optimized and lightweight as possible. In the future, more comparisons between the CED techniques should be done. We should continue the work of Björnsson et al. (2009) as one of the major challenges in the RCA of retrospectives is the high number of causes of problems. Similarly, we should continue to develop new emerging methods for capturing and refining the findings of software project retrospectives in order to improve the organizational learning. For example combing CED with retrospective timelines is an interesting future work area. We should also analyze the feasibility of software tools for the RCA of retrospectives. For example, software tools that support conducting RCA in distributed retrospectives are scarce (Lehtinen et al., 2014b).

Appendix A. Questions asked on Questionnaire 1

This inquiry is 100% anonymous. The people names won't be published. All the results are analyzed as a one mass of answers.

1 **Your name:** [. . .]

Answer by circling a choice for each question.

My role in the project team is. . . [1=project manager, 2=quality manager, 3=architect, 4=developer]

2 **Cause collection**
The scale was: [1=very bad; 2=bad, 3=somewhat bad, 4=neutral; 5=somewhat good, 6=good, 7=very good, ∗=I don't answer]

- Technique used to collect the causes is. . .   [Result ID = 1.3]
- Technique used to organize the causes is. . .   [Result ID = 2.1]
- Advantageousness of cause collection in comparison to used effort was. . .   [Result ID = 3.1]
- Correctness of the detected causes is. . .   [Result ID = 3.6]
- Easiness to solve the detected causes is. . .   [Result ID = 3.7]
- My effort in the cause collection was. . .   [Result ID = 4.3]
- Effort of my team in the cause collection was. . .   [Result ID = 4.2]
- Efficiency of the method to detect improvement targets compared to the other methods you have experience. . .   [Result ID = 3.2]

3 **General**
The scale was: [1=fully disagree; 2=disagree, 3=somewhat disagree, 4=neutral; 5=somewhat agree, 6=agree, 7=fully agree, ∗=I don't answer]

- There was an open communication in the session. . .   [Result ID = 4.1]
- In general, this was a useful workshop. . .   [Result ID = 3.4]
- The used RCA method helps to develop corrective actions. . .   [Result ID = 3.3]
- Team spirit of our project team is great. . .   [Result ID = 4.4]
- This workshop was nothing more than waste of time. . .   [Result ID = 3.5]

4 **General**
The scale was: [1=fully disagree; 2=disagree, 3=somewhat disagree, 4=neutral; 5=somewhat agree, 6=agree, 7=fully agree, ∗=I don't answer]

- Detecting the fundamental causes of the problem was challenging. . .   [Result ID = 1.1]
- Problem causes should be collected by writing them on papers. . .
- Problem causes should be collected by discussing on them. . .
- It is a good idea to articulate publicly the written causes. . .
- The participants purposefully did not name some important causes. . .   [Result ID = 4.5]
- The participants did not care to name all the causes publicly. . .   [Result ID = 4.6]
- The only way to solve a problem is through solving its fundamental causes. . .
- It was hard to me to get the big picture of the fundamental causes of the problem, because of their high number. . .   [Result ID = 2.2]
- It was easy to register the causes I detected among the other causes. . .   [Result ID = 2.3]
- It is important to collect sub causes of a problem. . .   [Result ID = 1.5]
- Technique used to collect problem causes is easy to use. . .   [Result ID = 1.2]
- Technique used to collect problem causes is useful. . .   [Result ID = 1.4]

Appendix B. Questions asked on Questionnaire 2

This inquiry is 100% anonymous. The people names won't be published. All the results are analyzed as a one mass of answers.

1 **Your name:** [. . .]
Answer by circling a choice for each question.

The scale was: [1=fully disagree; 2=disagree, 3=somewhat disagree, 4=neutral; 5=somewhat agree, 6=agree, 7=fully agree, ∗=I don't answer]

- I think that RCA should rather be conducted by using the directed graph than by using the structural list. . .
- The high number of causes in the first workshop created a problem of being difficult to get the big picture of the fundamental problem causes. . .
- It was easier in the first workshop to register the causes I detected among the other causes. . .
- Technique used to collect the causes in the first workshop is easier than the method used in this second workshop. . .
- Technique used to organize the causes in the first workshop is easier than the method used in this second workshop. . .
- The high number of causes in this second workshop created a problem of being difficult to get the big picture of the fundamental problem causes. . .
- It was easier in this second workshop to register the causes I detected among the other causes. . .
- It was easier to get the big picture of the fundamental causes of the problem in this second workshop than in the first workshop. . .
- Technique used to organize the causes in this second workshop is more difficult than the method used in the first workshop. . .
- Technique used to organize the causes in the first workshop is more difficult than the method used in this second workshop. . .

References

Ainsworth, S., Th Loizou, A., 2003. The effects of self-explaining when learning with text or diagrams. Cogn. Sci. 27 (4), 669–681.

Ammerman, M., 1998. The Root Cause Analysis Handbook: A Simplified Approach to Identifying, Correcting, and Reporting Workplace Errors, first ed. Productivity Press, New York, NY.

Anbari, F.T., Carayannis, E.G., Voetsch, R.J., 2008. Post-project reviews as a key project management competence. Technovation 28, 633–643.

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

Babb, J., Hoda, R., Norbjerg, J., 2014. Embedding reflection and learning into agile software development. IEEE Softw. 31 (4), 51–57.

Berander, P. 2004. Using students as subjects in requirements prioritization. Paper presented at International Symposium on Empirical Software Engineering, 2004. ISESE'04.

Birk, A., Dingsøyr, T., Stålhane, T., 2002. Postmortem: Never leave a project without it. IEEE Softw. 19 (3), 43–45.

Bjarnason, E., Hess, A., Svensson, R.B., Regnell, B., Doerr, J., 2014. Reflecting on evidence-based timelines. IEEE Softw. 31 (4), 37–43.

Bjørnson, F.O., Wang, A.I., Arisholm, E., 2009. Improving the effectiveness of root cause analysis in post mortem analysis: a controlled experiment. Inf. Softw. Technol. 51 (January (1)), 150–161.

Boh, W.F., Slaughter, S.A., Espinosa, A.J., 2007. Learning from experience in software development: a multilevel analysis. Manage. Sci. 53 (8), 1315–1331.

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

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

Carver, J., Jaccheri, L., Morasca, S., Shull, F., 2003. Issues in using students in empirical studies in software engineering education. Paper presented at Ninth International Software Metrics Symposium.

Cohen, J., 1988. Statistical Power Analysis for the Behavioral Science. Lawrence Erlbaum, Hillsdale, New Jersey.

Collier, B., DeMarco, T., Fearey, P., 1996. A defined process for project post mortem review. IEEE Softw. 13 (4), 65–72.

Dingsøyr, T., 2005. Postmortem reviews: purpose and approaches in software engineering. Inf. Softw. Technol. 47 (5), 293–303.

Dingsøyr, T., Moe, N.B., Nytrø, Ø., 2001. Augmenting experience reports with lightweight postmortem reviews. Paper presented at Third International Conference on Product Focused Software Process Improvement, 2001. PROFES '01.

Dybå, T., Maiden, N., Glass, R., 2014. The reflective software engineer: reflective practice. IEEE Softw. 31 (4), 32–36.

Flick, U., 2006. An Introduction to Qualitative Research. Sage.

Glass, R.L., 2002. Project retrospectives, and why they never happen. IEEE Softw. 19 (October (5)), 111–112.

Höst, M., Regnell, B., Wohlin, C., 2000. Using students as subjects—a comparative study of students and professionals in lead-time impact assessment. Empirical Softw. Eng. 5 (3), 201–214.

Ishikawa, K. (Ed.), 1990. Introduction to Quality Control. JUSE Press Ltd, Japan.

Jick, T.D., 1979. Mixing qualitative and quantitative methods: triangulation in action. Adm. Sci. Q. 24 (4), 602–611.

Juristo, N., Moreno, A.M., 2003. Basics of Software Engineering Experimentation. IBT Global, London.

Kampenes, V.B., Dybå, T., Hannay, J.E., IK Sjøberg, D., 2007. A systematic review of effect size in software engineering experiments. Inf. Softw. Technol. 49 (11), 1073–1086.

Larkin, J.H., Simon, H.A., 1987. Why a diagram is (sometimes) worth ten thousand words. Cogn. Sci. 11 (1), 65–100.

Latino, R.J., Latino, K.C. (Eds.), 2006. Root Cause Analysis: Improving Performance for Bottom-line Results, third ed. CRC Press, Boca Raton, FL.

Lee, S., Courtney, J.F., O'Keefe, R.M., 1992. A system for organizational learning using cognitive maps. OMEGA 20 (1), 23–36.

Lehtinen, T.O.A., Mäntylä, M.V., 2011. What are problem causes of software projects? Data of root cause analysis at four software companies. Paper presented at International Symposium on Empirical Software Engineering and Measurement, 2011. ESEM '11.

Lehtinen, T.O.A., Mäntylä, M.V., Vanhanen, J., 2011. Development and evaluation of a lightweight root cause analysis method (ARCA method) – field studies at four software companies. Inf. Softw. Technol. 53 (10), 1045–1061.

Lehtinen, T.O.A., Mäntylä, M.V., Vanhanen, J., Lassenius, C., Itkonen, J., 2014a. Perceived causes of software project failures – an analysis of their relationships. Inf. Softw. Technol. 56 (6), 623–643.

Lehtinen, T.O.A., Virtanen, R., Viljanen, J.O., Mäntylä, M.V., Lassenius, C., 2014b. A tool supporting root cause analysis for synchronous retrospectives in distributed software teams. Inf. Softw. Technol. 56 (4), 408–437.

Mayer, R.E., Gallini, J.K., 1990. When is an illustration worth ten thousand words?. J. Educ. Psychol. 82 (4), 715.

McLeod, L., MacDonell, S.G., 2011. Factors that affect software systems development project outcomes: a survey of research. ACM Comput. Surv. 43 (24), 24–55.

Moløkken-Østvold, K., Jørgensen, M., 2005. A comparison of software project overruns – flexible versus sequential development models. IEEE Trans. Softw. Eng. 31 (9), 754–766.

Moody, D.L., 2009. The "physics" of notations: toward a scientific basis for constructing visual notations in software engineering. IEEE Trans. Softw. Eng. 35 (6), 756–779.

Nakashima, T., Oyama, M., Hisada, H., Ishii, N., 1999. Analysis of software bug causes and its prevention. Inf. Softw. Technol. 41 (15), 1059–1068.

Ottensooser, A., Fekete, A., Reijers, H.A., Mendling, J., Menictas, C., 2012. Making sense of business process descriptions: an experimental comparison of graphical and textual notations. J. Syst. Softw. 85 (3), 596–606.

Rooney, J.J., Vanden Heuvel, L.N., 2004. Root cause analysis for beginners. Qual. Prog. 37 (August (7)), 45–53.

Runeson, P., 2003. Using students as experiment subjects—an analysis on graduate and freshmen student data, Paper presented at Seventh International Conference on Empirical Assessment in Software Engineering. Keele University, UK.

Runeson, P., Höst, M., 2008. Guidelines for conducting and reporting case study research in software engineering. Empirical Softw. Eng. 14 (December (2)), 131–164.

Schwaber, K., Sutherland, J., 2011. Scrum Guide. Scrum Alliance.

Siau, K., 2004. Informational and computational equivalence in comparing information modeling methods. J. Database Manage. 15 (1), 73–86.

Software Engineering Institute. CMMI for Development, Version 1.2. Pittsburg: Carnegie Mellon.

Stålhane, T., 2004. Root cause analysis and gap analysis – a tale of two methods, Paper presented at EuroSPI 2004. Trondheim, Norway.

Stålhane, T., Dingsøyr, T., Hanssen, G., Moe, N., 2003. Post mortem—an assessment of two approaches, Empirical Methods and Studies in Software Engineering, Leture Notes in Computer Science, vol. 2765. Springer, Berlin, Heidelberg, pp. 129–141.

Stevenson, W.J. (Ed.), 2005. Operations Management, eighth ed. McGraw-Hill/Irwin, New York.

Svahnberg, M., Aurum, A., Wohlin, C., 2008. Using students as subjects—an empirical evaluation. Paper presented at Second ACM-IEEE international symposium on Empirical software engineering and measurement.

Terzakis, J., 2011. Virtual retrospectives for geographically dispersed software teams. IEEE Softw. 28 (3), 12–15.

Vanhanen, J., Lehtinen, T.O.A., 2014. Software engineering problems encountered by capstone project teams. Int. J. Eng. Educ. 30 (6(A)), 1461–1475.

Vanhanen, J., Lehtinen, T.O.A., Lassenius, C., 2012. Teaching real-world software engineering through a capstone project course with industrial customers, Paper presented at First International Workshop on Software Engineering Education Based on Real-World Experiences, EduRex 2012. Zurich.

Vessey, I., 1991. Cognitive fit: a theory-based analysis of the graphs versus tables literature. Decis. Sci. 22 (2), 219–240.

Von Zedtwitz, M., 2002. Organizational learning through post-project reviews in R&D. R&D Manage. 32 (3), 255–268.

Yin, R.K. (Ed.), 1994. Case Study Research: Design and Methods, second ed. SAGE Publications, United States of America.

Timo O.A. Lehtinen received M.Sc. in software engineering from Aalto University, Espoo, Finland, in 2010. He is a researcher at Aalto University and his Ph.D. work focuses on root cause analysis, which he has studied both in the software industry and education contexts. He has years of software project management experiences in industry.

Mika V. Mäntylä is a professor at Aalto University. His current research interests include software defects, human cognition, and software evolution. He has a D.Sc. degree in software engineering from Aalto University.

Juha Itkonen is a postdoctoral researcher at Aalto University. His research focuses on experience-based and exploratory software testing and human issues in software engineering, including quality assurance in agile contexts. He received his D.Sc. degree in software engineering in 2012 from Aalto University.

Jari Vanhanen received D.Sc. in software engineering from Aalto University, Espoo, Finland, in 2011. He is a university lecturer in software engineering with Aalto University, and has been the responsible teacher for the capstone software development project course at Aalto University since 2001.