Information and Software Technology 56 (2014) 408–437

Contents lists available at ScienceDirect

Information and Software Technology

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

A tool supporting root cause analysis for synchronous retrospectives in distributed software teams

Timo O.A. Lehtinen *, Risto Virtanen, Juha O. Viljanen, Mika V. Mäntylä, Casper Lassenius

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

ARTICLE INFO

Article history:
Received 7 June 2013
Received in revised form 8 January 2014
Accepted 9 January 2014
Available online 17 January 2014

Keywords:
ARCA-tool
Root cause analysis
Distributed retrospective
Global software engineering

ABSTRACT

Context: Root cause analysis (RCA) is a useful practice for software project retrospectives, and is typically carried out in synchronous collocated face-to-face meetings. Conducting RCA with distributed teams is challenging, as face-to-face meetings are infeasible. Lack of adequate real-time tool support exacerbates this problem. Furthermore, there are no empirical studies on using RCA in synchronous retrospectives of geographically distributed teams.
Objective: This paper presents a real-time cloud-based software tool (ARCA-tool) we developed to support RCA in distributed teams and its initial empirical evaluation. The feasibility of using RCA with distributed teams is also evaluated.
Method: We compared our tool with 35 existing RCA software tools. We conducted field studies of four distributed agile software teams at two international software product companies. The teams conducted RCA collaboratively in synchronous retrospective meetings by using the tool we developed. We collected the data using observations, interviews and questionnaires.
Results: Comparison revealed that none of the existing 35 tools matched all the features of our ARCA-tool. The team members found ARCA-tool to be an essential part of their distributed retrospectives. They considered the software as efficient and very easy to learn and use. Additionally, the team members perceived RCA to be a vital part of the retrospectives. In contrast to the prior retrospective practices of the teams, the introduced RCA method was evaluated as efficient and easy to use.
Conclusion: RCA is a useful practice in synchronous distributed retrospectives. However, it requires software tool support for enabling real-time view and co-creation of a cause-effect diagram. ARCA-tool supports synchronous RCA, and includes support for logging problems and causes, problem prioritization, cause-effect diagramming, and logging of process improvement proposals. It enables conducting RCA in distributed retrospectives.

1. Introduction

Retrospectives, also known as post-mortems, are activities where the team members share experiences about problems and their causes [1], analyzing a recently ended project and/or iteration. Root cause analysis (RCA) is a structured investigation of a problem to detect which underlying causes need to be solved [2], and a useful practice for retrospectives [3–5]. Retrospectives are typically conducted in face-to-face meetings, in which the team members first identify problems that occurred. Subsequently, they conduct lightweight RCA by collaboratively creating a cause-effect diagram visualizing the causes of problems [5].

Global software engineering, employing geographically distributed teams, has become a standard way of operating in today's business [6]. This way of working creates new challenges related to geographical, temporal, cultural and organizational distance [7]. The use of distributed teams also creates a major challenge for conducting team retrospectives [8]. In previous work, we developed a lightweight focus group based RCA method, ARCA, and evaluated it in four industrial field studies using collocated teams [9]. Even though the method was well liked, the companies pointed out the need to conduct RCA with their distributed teams. Literature on distributed retrospectives identifies a similar need and discusses the use of a combination of email, spreadsheets and an online audio bridge to help facilitate the retrospectives [8]. However, relying on such tools in focus group based synchronous RCA is not feasible, as organizing and interpreting a high number of causes using emails and spreadsheets would be highly difficult. Instead, cause-effect diagrams [9] supporting real-time online environment should be used in distributed retrospectives.

* Corresponding author. Tel.: +358 407752781.
E-mail addresses: timo.o.lehtinen@aalto.fi (T.O.A. Lehtinen), risto.virtanen@aalto.fi (R. Virtanen), juha.o.viljanen@aalto.fi (J.O. Viljanen), mika.mantyla@aalto.fi (M.V. Mäntylä), casper.lassenius@aalto.fi (C. Lassenius).

0950-5849/$ - see front matter © 2014 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.infsof.2014.01.004

There are many proprietary software tools for RCA.[1] However, we have not succeeded in finding a web-based tool that fulfills the needs of conducting lightweight RCA in synchronous distributed software project retrospectives. First, the tool should make it possible for RCA participants to co-create a cause-effect diagram [5,9], which stays in-sync between the sites. Second, the tool should allow the development of process improvement ideas for the causes and maintain links between the improvement ideas and the detected causes [10–14]. Third, the tool should make it possible to vote on the most severe causes and best improvement ideas [9]. Fourth, the tool should also make it possible to capture and refine the findings of several retrospectives, in order to support organizational learning and knowledge management [3]. To the authors' best knowledge[2] the most frequently lacking feature of current software tools for RCA is the syncing mechanisms needed for simultaneous co-creation of cause-effect diagrams, see Table 1. There are tools for simultaneous graph drawing, e.g., Google Docs drawings [15], but these tools lack features to support RCA, e.g. automatically capturing and refining the findings of retrospectives.

Furthermore, to our knowledge, there are no empirical studies on the feasibility of using RCA in synchronous distributed retrospectives. While there is ample evidence for the benefits of RCA to detect the causes of problems and make improvements in various contexts [9–13,16–21], the existing studies have been conducted in a face-to-face context. Thus, in order to contribute to the existing studies, we developed an online tool for supporting synchronous RCA in distributed software project retrospectives called ARCA-tool.[3] It provides features for distributed RCA, idea development, and capturing the lessons learned in many retrospectives.

The goals of this paper are *to present ARCA-tool including its technology and main features, and to provide an empirical evaluation of the tool and synchronous RCA in the context of industrial software development with agile teams.* In order to evaluate the usefulness of RCA and ARCA-tool, we used interviews, questionnaires, and observations in the retrospectives of geographically distributed industrial software teams, that followed the Scrum methodology [22]. Our research questions were:

RQ1: Is ARCA-tool perceived as useful in the distributed retrospectives of agile software teams?
RQ2: Is ARCA-tool perceived as easy to use in the distributed retrospectives of agile software teams?
RQ3: Is RCA perceived as a good approach to use in the distributed retrospectives of agile software teams?

While the first two questions are related directly to ARCA-tool, we evaluate the RCA method, since the evaluators might have difficulty separating the effect of the tool and the context in which it was applied, i.e. the synchronous retrospective method used and the company context. Naturally, ARCA-tool can be used without the retrospective with the RCA method and vice versa.

The rest of the paper is structured in the following way. Section 2 covers the related work and identifies a gap in research, which is then filled by introducing ARCA-tool in Section 3. Section 4 explains the field study method used to evaluate the tool in real industrial contexts and the results of this evaluation are given in Section 5. Finally, Section 6 contains the discussion and Section 7 provides conclusions and directions for further work.

[1] http://open-tube.com/10-best-software-tools-to-conduct-root-cause-analysis-and-solve-complex-problems/.

[2] Investigation of proprietary RCA tools is difficult as freely available information of the tools is limited.

[3] http://wirca.soberit.hut.fi/prod/?language=en.

2. Related work

In this section, we introduce the concept of software project retrospectives and present problems related to conducting RCA with distributed software teams. We also compare RCA software tools that we have found.

2.1. Software project retrospectives

The key for effective problem prevention is controlling the causes of problems [23]. It is claimed that problems cannot be solved without solving their causes [9]. Retrospectives are one means to help identify and prevent the reoccurrence of problems that have occurred in prior projects [8,24–26].

In retrospectives, the team members share their experiences about problems and their causes [4,5,24]. Retrospectives enable learning at the individual, team, and organizational level. At the individual level, learning is based on shared experiences [27]. Thus, at the team level, learning is related to the shared experiences among the team members [27]. Furthermore, learning at the organizational level requires knowledge management, i.e. the shared experiences are captured and refined, and thereafter distributed to the teams [3]. Therefore, the output of retrospectives must be captured and refined.

A software project retrospective can be viewed as a step-by-step process [5,28]. In the first step, problems related to the past project, iteration, or milestone are identified. Thereafter, the participants collaboratively identify the causes of the problems by using RCA. In RCA, the causes of problems are identified by constantly asking "why" for every cause [9]. The causes are visualized by using a cause-effect diagram, e.g., a fishbone diagram [5,14,19], or a directed graph [5,9]. The diagram represents the cause-and-effect relationships between the causes of problems. It aims to assist the participants to detect underlying causes for the problems. After the cause-effect diagram is finalized, the participants detect the root causes, defined as the underlying and controllable causes of the problem [9]. Process improvement ideas are then developed for the selected root causes.

While the traditional use of retrospectives has been fraught with problems [25], modern agile development processes, such as Scrum [22], have made the practice common in modern organizations. As such, Scrum or other agile development processes do not require the use of RCA as part of their retrospectives – however RCA can well be used in Scrum retrospectives as a practice that helps add both structure and provides additional value to the teams.

2.2. Root cause analysis and distributed retrospectives

The issue of distributed team members has been considered as the greatest challenge that organizations face while conducting retrospectives [8]. Retrospectives should be lightweight [28] but under the influence of budget constraints and time pressure, they are rarely conducted [25]. While the project members are geographically dispersed, arranging face-to-face retrospectives requires too much effort. Conducting face-to-face retrospectives in such settings is often cumbersome. Distributed retrospectives are introduced as substitutes for face-to-face retrospectives [8]. Such retrospectives are typically conducted with the aid of an audio or video bridge [8]. Logically, in distributed software projects, conducting distributed retrospectives require less effort than conducting them face-to-face due to decreased traveling time.

Conducting RCA in distributed retrospectives is difficult as it requires tools that are not yet mature enough. It has been claimed that a combination of emails, spreadsheets, and an audio bridge are enough to support distributed retrospectives [8]. However, in