

Research Article

Team Exploratory Testing Sessions

Soili Saukkoriipi¹ and Ilkka Tervonen²

¹ R&D, F-Secure, 90580 Oulu, Finland

² Department of Information Processing Science, University of Oulu, 90014 Oulu, Finland

Correspondence should be addressed to Soili Saukkoriipi, soili.saukkoriipi@f-secure.com

Received 6 December 2011; Accepted 17 January 2012

Academic Editors: J. Fernandez-Ramil and H. Siy

Copyright © 2012 S. Saukkoriipi and I. Tervonen. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Exploratory testing (ET) is popular, especially among agile development teams. In this paper, we study the team aspect in the ET context and explore how to use ET in team sessions to complement other testing activities. The goal was to define a team exploratory testing (TET) session approach and to provide evidence that the approach is worth using. A TET session approach is defined by means of parameters, roles, and process. Also, instructions for using the approach are given. The team is the key factor that gives the approach its value and distinguishes it from basic ET. The team enables greater access to expertise, experience, and information. The TET session approach enables participants with different professional background to join in the session. The sessions may be focused on different purposes; they can contribute to finding defects or learning the system, for example. With careful parameter definition, the approach's risks are mitigated.

1. Introduction

This paper defines a new testing approach, a team exploratory testing (TET) session approach, and describes how to utilize it in a software project. The approach is based on exploratory testing (ET), and it fits well with agile testing but is not tied exclusively to it. The idea for the approach was obtained from ad hoc testing group testing sessions in which one of the authors participated. Basis for the research was Saukkoriipi's thesis *Defining and utilizing team exploratory testing sessions* (2010) mentored by Tervonen.

1.1. The Research Targets. Bach—among others—has noticed that ET has been studied little and not highly respected. Now the attitude is changing as companies seek more agile and cost-effective software development methods [1]. Textbooks, such as *Agile Testing* (2009) and *Exploratory Software Testing* (2009), are good indicators of this change. This paradigm was in the background when researchers wanted to study how to use ET in team sessions, and a project that the researcher worked for wanted to make its testing activities more effective.

The theoretical goal of this research was to *define a TET session approach* and describe how to utilize the approach in a software project. The idea of *team ET sessions* is shortly mentioned by Bach, 2003 [1]:

Team exploratory testing can be extremely powerful. In the experience of many test leads who've tried it, the social energy of people working together, hunting for bugs on the same equipment at the same time, often leads to more and better ideas than would otherwise come out if the same people worked independently. One way to organize team ET is to put testers into pairs and have them share one computer as they test. Another way I've done it is to have one tester "drive" at the keyboard while several others watch and comment. If the driving tester discovers a problem or has a question that needs to be researched, one of the watchers can break away to attempt to investigate that issue using another test platform. That frees the driving tester to continue the main thread of testing with less distraction. This method works especially

useful as a blockbusting tool to get the test effort out of a rut or to help train testers in the technology of the product or about methods of test design.

Before this paper, the idea is not developed further in the literature. This paper describes the approach, parameters that enable TET session, TET session roles, and overall TET session process. It also describes how to use the approach in practice.

In addition to the research targets, the study had practical goals that the target project set. In the target project, the reasons to utilize ET were the following characteristics: agility, cost-effectiveness, and the ability to enrich traditional test-case-based testing (TCT). During the research, it was desired for ET to be utilized more efficiently using TET sessions. It is important to note that TET sessions did not replace any other testing activities in the project and that comparing the TET sessions and other testing results was not within the scope of this research.

1.2. Exploratory Testing. Exploratory testing is simultaneous learning, test design, and test execution [1]. It is done in iterations: (1) the tester studies the system being tested; (2) based on her experience and knowledge, the tester makes guesses about where defects or some other interesting behavior might occur; (3) the tester designs a test to reveal possible defects or other interesting aspects of the tested software; (4) the tester executes the test and observes carefully to determine whether the guessed defect or some other strange behavior takes place; (5) the tester reflects on the meaning of her observations and then, based on the learning just gained, starts from the beginning. The session duration varies from an hour to three hours. It is important that iterations are executed several times to ensure the learning that is essential in finding defects [2].

Exploratory testing has been addressed in software testing textbooks since the 1970s, (e.g., [3]) although it has been treated as error guessing or an ad hoc method [4]. SWEBOK 2004 separated ET from ad hoc testing and error guessing [5]. According to Bach [1], ad hoc often is synonymous with careless work, and the term “ET” is preferred by test methodologists. Kaner was the first to use the term “exploratory testing” in 1983 [6].

ET can be applied to any test technique [6, 7]. It can be utilized in designing new test cases or analyzing already executed test runs. It can be performed both manually and using test automation [6]. ET suits all development phases and, due to its interactivity and reliance on feedback, it is particularly suited for agile development [8]. According to Bach [1], ET fits the following situations particularly well:

- (i) Learning a product: learning requires exploring and questioning [9].
- (ii) Getting rapid feedback on a new product or a feature. ET is an especially efficient way to test a product quickly when starting from scratch [10].
- (iii) You have already tested using predefined test cases and seek variation.

- (iv) Investigating and isolating a particular defect. Or when you try to find the single most important defect in very limited time.
- (v) Checking the work of another tester by doing a brief independent investigation.
- (vi) Investigating the status of a particular risk to evaluate the need for scripted test cases in that area.
- (vii) Improvising on scripted test cases, improving existing test cases, and writing new test scripts. The process of writing test scripts is exploratory [9].
- (viii) Interpreting vague test instructions, product analysis, and test planning.
- (ix) Testing based on reading the user manual and checking each assertion.
- (x) Regression testing based on old defect reports [1]. It is quite common that defect fixes succeed only partly or cause new defects. ET allows tester to look the tested features as whole, and holistic approach eases problem recognition [11].

Exploratory testing has some similarities with software inspection, which has a long history in the software engineering area. Fagan’s early paper [12] in 1976 explained the planning, measurement, and control functions in design and code inspections. Later, different types of inspection are introduced, such as structured walkthrough [13], peer reviews [14], perspective-based inspection [15], and perspective-based reading [16]. The major goal in ET and software inspection is to discover defects and do it with team working, for example. The weights of phases in ET and software inspection are different; the participant preparation phase, which is the most important for successful inspection, is missing in ET.

The benefits of ET include the ability to increase the effectiveness of testing, simultaneous learning, the ability to minimize preparation documentation, the ability to perform without comprehensive documentation, and rapid feedback flow [11]. It is significant that ET can be continuously optimized throughout the test process [1]. ET adapts to the situation and, even with severe time pressure or poor documentation, it can be effective [2].

Among ET’s disadvantages are the following: ET’s test coverage is very hard to determine if documentation is neglected [2], tracking the progress of individual testers or the testing as whole is difficult, and following up what has been tested is challenging [11]. ET’s effectiveness depends on many factors, such as the maturity of the software, the skills of the tester, the tested product itself, and the time available for testing [17].

Bach stated that ET’s power relies on information flow; new tests are done based on feedback from recently executed test cases. If this information flow is weak for some reason, it is better to move to prescribed test cases. Usually, the most effective testing is done using a well-considered combination of ET and TCT. “Exploratory testing is not against the idea of scripting” [1]. Each type of testing feeds into the other [18]. If you blend ET and TCT, you get the advantages of

both approaches without suffering from the disadvantages of either [2]. TET sessions use ET as the common working method.

1.3. The Research Method—Action Research and Design Science. In this work, two types of research methods were applied: design science and action research. In the research, these methods are intertwined so that design science guides the definition and evaluation of the TET session approach as a whole, and action research refines and guides the completion of research cycles. With the action research method, executions of TET sessions could be followed up in a controlled way.

Action research is a cyclic process consisting of five different phases: (1) diagnosing, (2) action planning, (3) taking action, (4) evaluating, and (5) specifying learning [19]. The approach is focused on and involved in the change process [20]. It requires the establishment of a client-system infrastructure or research environment that describes the boundaries of the research domain and the entry and exit of the researchers. Also, it defines the responsibilities of the client and the researcher(s) to each other. It involves the close collaboration of both researchers and practitioners [21]. Action research is an iterative process involving researchers and practitioners working together on a particular cycle of activities, including problem diagnosis, action intervention, and reflective learning [22]. In this research, the researcher's role as the facilitator in TET sessions made these activities possible.

Although action research is a combination of research and practice, it produces scientific findings. Through action research, problems in concrete settings are represented in a generalized form. This enables action research to serve both theory and practice. Knowledge of the scientific community can benefit from theoretical elements that come up during the action research [21]. Also, design science is inherently a problem-solving process and seeks innovative solutions intended to solve identified organizational problems [23].

Successful action research has three distinctive characteristics: the researcher is actively involved, new knowledge was gained during the research and can be immediately applied, and the research is a cyclical process linking theory and practice. The challenges are a lack of impartiality, lack of discipline, confusion with consulting, and context-bound nature. These problems are not specific to action research and can also be faced using other methods. To avoid these risks, diagnosis must be carefully based on the theory, and data collection must be systematic. Reports must allow future work to confirm or refute any claims of generalized theory [21]. A research diary, interviews, and questionnaires were a part of this research. Each action research phase, observations, and results were collected via a diary and notes. Interviews and questionnaires were used as channels of feedback and ideas. Sections 2.1–2.3 introduce the research data.

For the design science research to be completed, the guidelines presented in Table 1 should be addressed [23].

1.4. Structure. This paper is divided into sections as follows: Section 2 describes research settings and describes the flow of

the research to provide evidence for the approach. Sections 3, 4, 5, and 6 introduce theoretical results of the research: 3 the TET session approach, 4 roles, 5 parameters and related risks, and 6 process. Section 7 describes how to utilize the approach in a software project. Section 8 reflects the results gained during this research with respect to the target project's goals. Threats to the research validity are discussed in Section 9, and Section 10 gives the final conclusion of the research.

2. The Research Settings and Iterations

The target project was conducted by a service-focused organization at a big telecommunications company. The organization utilized Scrum as its software development method. The software under development was the Web Runtime (WRT) widget, a Web application focused on social connections and written in Java by around seven developers. In addition to the Scrum team, there was a testing team whose members participated in development's daily Scrum, planning, review, and retrospective meetings. The surrounding organization had several supporting roles, such as UI designers. The researcher worked as a senior test engineer in the target project, and the other writer of this paper tutored her in the research.

During the target project's development process, several software testing stages and types were used. The main ones were unit testing, integration testing, functional testing, nonfunctional testing, regression testing, and usability testing. The developers used a test-driven development (TDD) approach and utilized peer reviews, definition of done checklist, and static analyzers. ET was utilized by both the development and testing teams for the following purposes: learning the target software, test planning, improving existing scripted test cases, defect report-based retesting, emulator testing, and smoke testing. TET sessions were not arranged, but the surrounding organization had some group ad hoc testing experience. Ad hoc testing is based on tester's skills, intuition, and experience [5], but it is also unsystematic, not disciplined, and not repeatable approach [1].

The project's schedule was tight: it started in 2Q 2009, and its code was completed in early 1Q 2010. In addition to the tight schedule, requirements changed often, and documentation was limited. Due to these pressures, new ways to improve testing efficiency were explored, and it was decided to try TET sessions because the researchers saw potential in group ad hoc testing sessions in which the researcher had participated in and project's experience with basic ET was good.

The TET sessions were intended to *gather feedback and enhancement ideas* for the development. The application under development was new, and the target project had quite a free hand with the implementation details because no elaborated requirements were available. The testing team was encouraged to provide feedback to the development team; in addition to defect reports, enhancement ideas were welcomed. TET sessions were thought to be a good channel for gathering feedback and ideas. Usually, ET provides all kinds of information about a tested system, not just defect

TABLE 1: Design science guidelines and how they were addressed in this study.

Guideline	Description
(1) Design as an artefact	A viable construct, model, method, or instantiation must be produced. In this study, a TET session approach was to be defined and further refined in a software project.
(2) Problem relevance	An unsolved, important, and relevant business problem must be addressed. The value of the ET is recognized, but companies do not know how to conduct the ET approach using teams in a controlled way and what the benefits and obstacles of the TET session approach are. The quality of the artefact must be demonstrated.
(3) Design evaluation	In this study, the TET session approach has been used in a real software project, and research data was collected in the form of interviews, questionnaire answers, a research diary, and approach utilization results.
(4) Research contributions	Clear and verifiable contributions must be identified. This study introduces the TET session approach, which merges team and ET and provides a solution to improvement areas exposed in the testing process of the target project.
(5) Research rigor	Construction and evaluation of the design artefact must be carried out scientifically. In this study, three research cycles were conducted, following the action research method. During each cycle, research data was collected and, based on the analysis, new refinements of the approach were introduced.
(6) Design as a search process	The artefact is created in an iterative process until it satisfies the problem environment. During the research, feedback for the TET session approach was gathered from the target project in three research cycles, and the approach was refined in accordance with the feedback.

reports, and is, therefore, an effective way to gather and provide feedback [8].

Also, *new perspectives on the testing* were wanted. ET is not meant for exhaustive testing but for adding another dimension to the testing [24]. The aim was to *complement other testing activities*. The ET approach produces different types of test cases: “ET is primarily creative. It has value in terms of the problems we find while doing it, but it also has value because it is a way to create other kinds of tests” [25]. In the target project, TET sessions were used to get the target software tested from various viewpoints, boosted by the teamwork synergy. Further, the TET session findings were used in test case creation, updating, and weak test area recognition. ET often reveals areas that could use more tests [24].

Because ET is adaptive and often reveals quite unique defects [8], an important TET session target was *finding leaked defects*. It was desired that defects leaked from the other testing activities be found as early as possible to save effort and costs. Getting things right early is less expensive also when developing a small and noncritical system, as in the target project. For this kind of system, according to Boehm and Basili [26], finding and fixing defects after delivery is approximately five times more expensive than doing it before the delivery.

The last target was *social effects and learning*. In the target project, all of the personnel did not know each other, and there were also new beginners. Members of work groups usually continue to relate to one another also after the group task is completed [27]. In the TET sessions, people connected to the target software were able to meet each other and, in addition to learning the target software, grow their professional network and, thus, improve daily work efficiency. Learning the software and new testing skills was an important target since the testing team, in particular, had

several new testers and the learning pace was expected to be fast. Exploratory learning encourages learning through exploration in order to gain skills and knowledge [28, 29]. Especially for novices, exploratory learning is an effective way to acquire important knowledge about new systems [30].

The research started in September 2009. During the research, three action research iterations were executed. The first iteration was the largest; during it, the TET session approach was described and, during the following iterations, it was reviewed and polished. The research approach was direct, as the researcher was arranging and leading the sessions. The entry criteria for the research were the approved plan, while the exit criteria were three executed and reported TET sessions. Figure 1 illustrates how the research progressed.

The research data was collected by utilizing two interviews, three questionnaire rounds, and observing and taking notes during TET sessions. The research diary was maintained, and many informal conversations were held during the research. The facilitator and one participant of the organization’s earlier group ad hoc testing sessions were interviewed. All TET session participants were sent questionnaires after the realized sessions. Below the main phases and results of each research iteration are described.

2.1. First Iteration—Defining the Approach. The first action research iteration and its diagnosis phase started by exploring the target project’s motivation to organize TET sessions. The approach utilization targets were already introduced in Section 2. The quantitative TET session results that the target project was interested in were defects, enhancement ideas, test case improvements, and recognition of low test coverage/problematic coding areas.

An important source for the planning was feedback from the earlier group ad hoc testing sessions, arranged

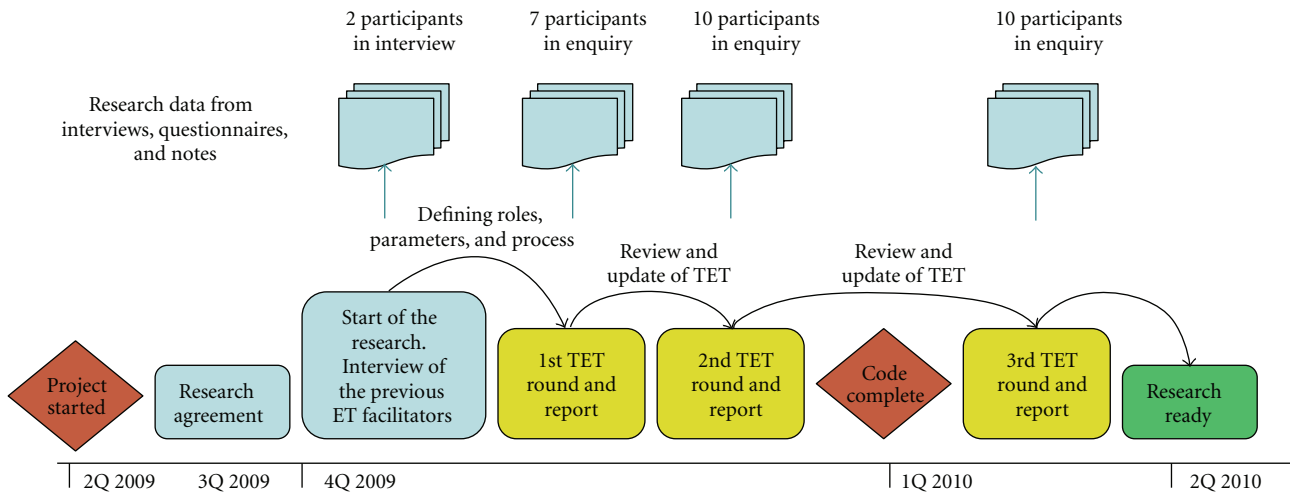


FIGURE 1: Research overview and timeline.

at the same organization, whose facilitator and participant were interviewed. The interview approach was qualitative interviews focusing on the interviewee's point of view and allowing interviewees to describe what is meaningful to them using their own words [31].

The interviews revealed that, during the earlier group ad hoc testing sessions, no process or clear roles were utilized and no reporting was done. Because of these lacking factors, the approach was impossible to repeat, and its benefits were hard to judge. Still, those sessions offered several important learning points, which were taken into account when planning the new TET sessions.

- (i) The results are not guaranteed or achieved easily. This has to be taken into account, and the risk can be minimized by preparing carefully.
- (ii) It is good to keep the team size at around ten participants so that the sessions stay manageable, and communication is easier.
- (iii) It is hard to get busy people to participate, so practical arrangements starting with a proper invitation and the sessions' point in time in the development cycle have to be carefully considered.
- (iv) It is beneficial to invite people with different backgrounds and skills to participate. This leads to a richer variety of defects. All do not have to be testers, but basic knowledge about the tested software is good to have.
- (v) A session has to start with an introduction to the tested software and the session itself. Thus, the session gets focused.
- (vi) A session has to be gently controlled to avoid unwanted meandering and getting off track. It is important to stay focused and avoid chasing defects that might not be so important to what is currently tested [24].

- (vii) Arranging sessions requires a lot of effort and is time consuming.

During the action planning phase, the concept of the TET sessions was roughly formulated, and TET session parameters were identified and described. Because testing is always context dependent [32], the parameters have to be valued before any action can be taken. In addition, the roles and responsibilities and the overall process were described to carry out successful TET sessions. The definitions were based on ET theory, the researcher's group ad hoc testing session experiences, and feedback from the interviews. The definitions were preliminary and, during the research iterations, they were tested in practice, evaluated, and developed further.

In week 40 of 2009, the first TET session, that is, action taking, took place. The researcher acted as the TET facilitator, and the target project's Scrum master handled the TET domain expert's role. Among the five TET participants, two were project testers, one was the project's test team leader, one was the project's error manager, and the last was the project's developer. Only a couple of the participants had heard about ET before.

For the first session, 30 findings were reported. Among these, 25 were defects and five were enhancement ideas. After the check, 11 new defects were reported further to the error management tool, and one idea was reported to the sprint planning tool. Altogether, the findings caused 21 test case updates: seven existing cases were updated, and 14 new test cases were created. Based on the defect clustering, one area was recognized as having weak testing coverage and possible coding problems.

During the analysis phase, the results, session feedback, and other observations were reflected against the goals that the target team had defined for the approach. The TET session approach, in general, was also considered. The results of the analysis are described in detail in Section 8; this applies also to the other iteration introductions.

Four learning points were recognized during the iteration: that the use of TET session supports other testing activities, that arranging TET sessions requires a lot of effort, the importance of sufficient participants, and the importance of focused testing.

The topic of this research was not a comparison of the effectiveness of ET and TCT, to replace the target project's TCT practices with ET, or to replace ET practices with TET sessions. During the research, the target project's ways of utilizing ET was enhanced with TET sessions, and the quantitative results were plentiful, but individual ET and other testing methods also produced results. This indicates that TET sessions can be used to complement both TCT and individual ET as well.

Arranging a TET session required considerable effort from the session facilitator. The preparation took around ten working hours and the completion approximately three working days, in addition to the two hours required by the actual session. This has to be taken into account when planning TET sessions since both preparation and completion have a big impact on the results and, therefore, must not be underestimated.

Choosing participants is a major factor to consider. During the session, the researcher noticed that some participants made lots of findings and some not at all. This phenomenon was not explained, for example, by long testing experience. Attitude—eagerness to learn the target software thoroughly, finding its weak parts, and using one's own and the team's creativity in doing so—seemed to be the most important factor. The error manager used his defect knowledge and found several new defects. The test team leader, who was part of the target project's management, reported many defects, asked good questions, and did not stiffen the atmosphere at all. This confirmed that management representatives can be good participant candidates.

The importance of focused testing came up with the defect reports that could not be reproduced afterwards, as the steps were insufficient. At all times during the testing, the testers must, in some way, record what they are doing to recognize defect root causes because, if the findings cannot be reproduced, they have no value. Even though ET is free in nature, it has to be focused and controlled by the practitioner. Proper reporting templates and good session preparation and introductions help, as do the facilitator's and domain expert's support.

2.2. Second Iteration—Polishing the Approach. During the second iteration's TET analyzing and planning phases, the results and feedback from the first iteration were examined. Several TET session process steps and a couple of parameters were added. Based on the feedback, instead of testing the whole application, some specific focus areas could be used, so it was decided to focus on security in the second TET session. Security testing was selected because little had been done earlier and security-related defects are usually critical and have to be corrected right away.

In week 46 of 2009, the second session took place. It was six weeks, three sprints, after the earlier session, and lots of fixes and new features were implemented in the meantime.

Eight persons, excluding the facilitator and domain expert, participated. Three of them were security testing experts. One of the participants had just started working as a project test engineer, and he was invited to learn the target software as well as provide a fresh viewpoint. The other participants were two developers, a UI specialist, and another project test manager. In the participant selection, testing experience was emphasized because of the focus area and because big architectural changes had been made, and the application usability was bad.

Altogether, 26 findings were reported. Among these, 25 were defects, and one was an enhancement idea. As a result, ten new defects and one enhancement idea were reported. Altogether, the findings caused 11 test case updates: seven existing cases were updated and four new test cases were created. At this time, no particular weak testing/coding area was recognized. Going through the results again took approximately three working days. Routines cannot significantly shorten the examination of test reports since each report is different.

The following learning points were recognized or strengthened during this second iteration: software condition and teamwork skills must be taken into account when selecting participants.

The software condition has a big effect on TET session testing, results, and even the atmosphere. The test manager, who had good basic knowledge of testing but limited hands-on ET experience, found the immature software difficult to test; there was confusion, and testing was limited. One of the participants found the experience to be "perhaps beneficial for the dev team as multiple previously unknown errors were found. However, the sw solutions were perhaps a bit too immature at time being." Still, the danger that the TET session would turn to aimless wandering if testers lost motivation as a result of poor software condition was avoided and, with experienced testers, results were gained. When selecting participants, the software maturity and the participants' testing skills should be in balance; for unstable software, it is better to invite only experienced testers. This finding supports Tuomikoski's [17] statement that low-skilled test engineers are at their best with mature software and high-skilled testers achieve good results regardless of the tested software's condition.

This session revealed also that teamwork skills should be carefully considered when selecting participants. Some of the participants were quite loud and made jokes about the application and its performance. It did not have a good effect on atmosphere, and one of the participants commented on this in his feedback. Some people have little competence in working collaboratively with other people and even one or two such individuals can significantly impede the ability of a team to utilize its members' expertise effectively [27].

2.3. Third Iteration—Finishing the Approach. During the diagnosis and planning phases, the researchers examined the results and feedback of the earlier rounds and continued reviewing and polishing the approach. As a result of this last

iteration, final versions of the approach, roles, process, and parameters were written. The iteration phases were also used as a basis for modeling how to use the approach in practice.

The third session was organized in week 11 of 2010, when the code was complete. Thus, the session was planned to focus on finding defects. The release date was drawing closer, so there was no reason to discover new features. The session was rather successful, and more defects were discovered than during earlier sessions. Ten people participated, excluding the facilitator and domain expert. The participants were already earlier participants as error manager, the target project's UI designer, the project's three new test engineers, and two senior test engineers—other of them outside the target project to get a fresh view.

Preparation for the third TET session took one working day. Approximately two working days were used to check the findings. The one-day improvement was caused by the better reports and by focusing on defect finding and passing on the improvement ideas. The improvement in the reports was the result of an updated defect template and a more well-established approach. The presence of the error manager helped again to recognize already-reported defects.

33 findings were reported and, after the check, 14 new defects were recorded, two of them are critical. Four test cases were updated, and six new cases were created. No particular weak testing/coding area was recognized.

The most important learning point from the session was the TET session approach's high potential to be used in new test engineers' tutoring process. Three new test engineers participated, and their feedback was encouraging; the session was used to learn the target software and to get acquainted with other team members.

3. TET Session Approach

The TET session approach consists of three parts: the team, already discussed ET, and the session. In this section, these terms are first defined individually and, in the end, the terms are combined, and an explanation of the TET session approach is given.

3.1. The Team. The team is a small number of people with complementary skills who are committed to a common purpose, performance goals, and approach, for which they hold themselves mutually accountable [33]. It is, thus, essential that the team members have different skills and backgrounds and that they act in a specific role and strive for common output. A group of people is a team if it is recognized as a team by nonmembers, it demands peak performance from all its members, it shares its rewards, and it is small in size. However, in the real world, only a few teams satisfy all these characteristics [34].

The teamwork benefits are the different viewpoints and knowledge bases that usually produce more solutions than an individual person would bring. Carmel [34] stated:

A good team encompasses a set of benefits that is seductive to any organization. It creates synergistic ideas and innovations; it is better

at objectively evaluating ideas and finding mistakes; it provides greater access to expertise and experience; it enhances motivation and commitment to the task; it has greater access to information; and it is a very flexible unit of work.

It is no surprise that, for many software development projects, teamwork is extremely important and believed to be crucial for the success of a project [31]. To design a productive team, the next conditions should be met: a team task is challenging, important to team's organization or customers, owned by the team, and consequential for the team members. Team designing has to be done well, otherwise the result can be waste of time and energy of members [27].

The team is the core factor that separates the TET session approach from basic ET. A TET team member, a participant, does not have to have a testing background, and she can participate in a session without any preparations. What is required is that she has experience or information that others might not have, is willing to share that knowledge during the session, and works towards the session goal. It is essential that TET session team members feel the session goal challenging, consider session a meaningful piece of work with a visible outcome that makes a difference, feel that they are responsible for the outcome, and have autonomy for deciding about how they do the work. The facilitator takes care of the session practicalities and ensures that results reported forward, like defects, are high quality.

The TET session approach has some similarities with inspections and the black team testing method. It is important to find the right people both in inspections and in TET sessions so that many viewpoints will be covered. In particular, this aspect is emphasized in perspective-based inspections [16]. Inspections and TET sessions can have also similar goals like finding defects, but unlike inspections, TET sessions represent dynamic testing and do not require any preparation from the participants. The black team method places an emphasis on team spirit; that is, the team take pride in what they are doing, are dedicated to its goal, and feel that they are a part of something unique [35]; successful TET sessions have the same qualities.

3.2. The Session. In this research, the term "session" is used with the same meaning as in a session-based test management (SBTM) approach. Main difference between TET session and SBTM approaches is that the former is utilized by teams, the latter by individuals. SBTM approach helps in tracking individual testers' ET progress [36]. SBTM gives a framework to a tester's ET so that results can be reported in a consistent, accountable way [24]. SBTM is a technique for managing and controlling unscripted tests, and it enables unscripted testing to become a strong part of an overall test strategy. The technique is based on the idea of effective limits. A test session has limited duration and, during a test session, a tester explores a limited part of the target system. Within these limits, activities are not controlled but left to the tester's judgment. The aim of these

limits is to focus the tester's attention, increase the reliability of metrics and the repeatability of tests, and limit the cost of poor exploration. The tester records her activities, including the reactions of the system, data used, conditions, diagnosis, or ideas [37].

Similarly, a TET session is a testing work unit; it is an uninterrupted block of reviewable and chartered test effort. "Uninterrupted" means that testing is done without interruptions such as emails or telephone calls. A session is a protected block of time allocated for focused testing [38]. "Reviewable" means that, as a result of the session, a report is produced, and it provides all the needed information to interested parties, such as test management. "Chartered" means that each testing session has a mission: it dictates what is tested or what kind of problems are looked for [36]. During a session, testing is focused on something specific, such as a certain feature, characteristic, or business scenario [37]. Also, the session structure is similar to that of SBTM; sessions are divided into three tasks: session setup, test design and execution, and result investigation and reporting [24].

3.3. TET Session Approach. According to Bach, team ET can be extremely powerful. In team ET, session participants work together hunting for defects using the same equipment at the same time. This leads to social energy and more and better ideas than working independently. Bach referred to situations in which people, for example, work in pairs and share test equipment, or one tester executes tests while others watch and comment [1].

In this research, a TET session is an uninterrupted, reviewable, chartered, and focused team testing work unit in which a carefully selected group of people do not share test equipment or watch one tester executing tests as in Bach's example; rather, all test with their own devices together as a team. With own devices, it is ensured that all the participants have equal possibility to test and do it exactly in their own way. Pairing is, of course, not forbidden, but the main way of working is testing together with the others doing the same and sharing experiences. The session has a facilitator who keeps it focused by setting the targets and ensuring that all participants have all the necessary information and tools.

TET session testing is done simultaneously in the same room, with the same version of the target software, and with a similar approach and equipment. The team shares a common goal, such as identifying defects. ET serves as the common working method, and any tool or technique can be used. The participants have different skills and backgrounds to enrich the communication and learning that are necessary for ET. The session goal and testing focus area are specified before the session. The TET facilitator leads the sessions. Quantitative session work products such as defect reports are treated as products of the whole team, not individuals. TET sessions are organized regularly and have regular participants, so the work can be seen as a constant practice.

TET sessions fulfill team characteristics. The sessions have a clear target set before each session, such as finding defects. The TET session team has three regular members: the session facilitator, domain expert, and domain management.

The participant role is also regular, but the people representing the role change. The approach and domain introductions and the facilitator's and domain expert's support keep the sessions focused and the participants informed about what to do and how to do it. The facilitator keeps the process up to date and uses participant feedback in doing so; thus, the team controls the approach.

Carmel [34] stated that an ad hoc collection of individuals is just a group, not a team. TET sessions avoid this threat with its regular members and because the team participants are not collected without considering their skills and backgrounds, and usually, most of them are invited from the same organization for which the testing is done.

The TET session approach is a framework for managing and controlling team ET. The approach does not concentrate only on the session meeting but has equally important, specific phases before and after the session. The purpose of the approach is not to stiffen ET but to give sessions a strong skeleton so that the team can concentrate on the essentials: meeting the session goals, sharing experiences and knowledge, and learning. In the kernel of the approach is the team that gathers to do ET. Work is prepared, controlled, and finished using the TET session approach. How the approach is built up is illustrated in Figure 2.

4. TET Session Roles

The team design should lessen the chances that members will encounter build-in obstacles to good performance. A good design creates conditions that make it easier and more natural for task-effective behaviors to emerge and persist [27]. The following TET session roles and responsibilities were designed to form a TET session team structure, task structure, and team norms during this research. There might also be other interested parties, but they are not associated with any session-related actions and, therefore, are not counted as a specific role.

4.1. TET Facilitator. The TET facilitator defines how the TET session approach is utilized in the target project and adjusts the process for the target project's needs and keeps it up to date.

The facilitator arranges TET sessions and takes care of the session parameter definitions and process execution. For example, she selects and invites the participants, arranges the facilities, takes care of the test environments and tools, acts in a supportive role during sessions, handles result reporting, and collects feedback. If there is time during sessions, the facilitator can act as a TET participant.

The person acting as the TET facilitator has to know the target domain and its testing as well as the TET session approach. She also has to be able to lead a team. Possible candidates for TET facilitator role are, for example, senior test engineers. Because the TET facilitator is responsible for the approach utilization, her role should be permanent.

4.2. TET Domain Expert. The TET domain expert introduces target software (version) to TET participants and supports its usage during sessions. She comments defect

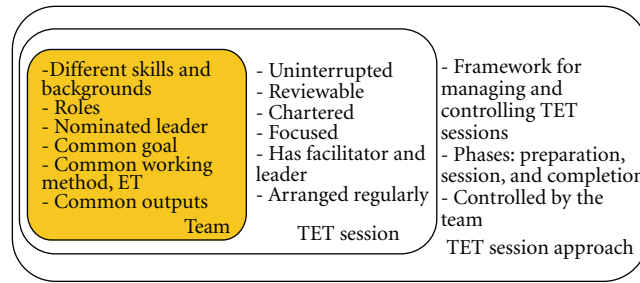


FIGURE 2: The team ET session approach.

and enhancement idea candidates—for example, she tells whether something is planned or reported already, though commenting has to be done in a subtle way to not discourage anyone. If there is time during sessions, domain expert can act as one of TET participants. After sessions, she acts as a source for the TET session approach feedback.

The TET domain expert should know the target domain thoroughly and be able to present domain overview briefly and understandable. Commenting findings require diplomatic skills. Possible candidates are Scrum masters and other specialists. Together with the facilitator, the TET domain expert is enabling TET session success and to be handled efficiently role should be permanent.

4.3. TET Participant. The TET participant allocates time and attends the TET sessions. She does ET, shares information, and reports her possible findings during sessions. After a session, she acts as a source for feedback on the TET session approach.

Participants are selected mostly from the target organization's personnel. The natural TET participant candidates are test engineers and design engineers who do testing as a part of their daily jobs. Because the TET session aim is to look at the target application from different viewpoints and share knowledge, people familiar with the target software are also good candidates and, for them, a lack of testing experience is not a reason to be excluded. The aim is to form heterogeneous teams to enable information sharing.

To keep sessions informal, management participants have to be considered carefully. If managers are not too formal, it could be beneficial to get them to participate. Managers have big-picture knowledge about the target system, and by participating, they get valuable hands-on experience. For example, test managers can be considered as possible participants. According to Kaner et al., test managers, if they do not work on projects, will lose many technical skills over time and, as a result, not see the real problems that their testers are coping with, find it harder to judge the quality of their staff's work, and may even lose credibility among staff. The test manager has to have enough knowledge about the target domain and its testing in practice that she can talk intelligently about it with her staff [38]. A TET session is an excellent place to get in touch with the target software, its testing, and its latest challenges in practice.

4.4. TET Domain Management. The TET domain management allocates time for TET sessions and pays for other possible expenses. They follow up on the approach utilization results, can give feedback for approach-improvement purposes, and are reported after sessions have taken place. They should be familiar with the TET session approach to be able to nominate right people to facilitator and domain expert roles, to judge its results and decide whether more sessions should be arranged. Hackman [27] puts it well: "A manager who wants a team task to be done well cannot simply call some people together, toss them a task, and hope for the best."

The domain management does not participate in TET sessions. Individual management members might participate, but they do so as TET participants and are invited by the TET facilitator.

5. TET Session Parameters and Related Risks

TET session parameters are characteristics that have to be valued before each session. Some of the parameters are more stable than the others; for example, frequency and duration can be adjusted after the first session, but it is not necessary. The participants and software version, on the other hand, have to be defined before each session.

TET sessions require resources, so they should also produce the best results possible, and working parameters are the key elements. Well-chosen parameter values prevent approach-related risk exposure. At the end of this section, risks are discussed.

5.1. Goal of the Session. The goal of the session is the most important approach parameter, as it has a big impact on the other parameters. For example, if the goal is to learn a new testing technique, other parameters have to reflect this target. By defining context-fitting goals, the approach is utilized most efficiently. The possible TET session goals are as follows.

- (i) Finding defects not found by other testing practices.
- (ii) Producing enhancement ideas.
- (iii) Complementing TCT with new perspectives and new and updated test cases. Test case creation and updating is based on findings done during TET sessions.

- (iv) Recognizing poor test coverage/coding areas. This is done by analyzing session results; if findings cumulate in a particular area, it might be a sign of weak testing, coding, or both.
- (v) Quickly assessing the target system's quality level, such as for a certain milestone.
- (vi) Learning the target system. The session is a great learning opportunity due to the presence of other participants and domain expert.
- (vii) Learning new testing skills: the TET session approach, ET, and, for some, testing in general might be a new experience.
- (viii) Learning a new test tool.
- (ix) Learning a new test technique.

The team aspect also makes the following goals possible.

- (i) Giving live feedback for the development. The presence of domain expert and possible development team members makes this possible.
- (ii) Sharing knowledge with others with different backgrounds and skills but still connected to the target software.
- (iii) Getting variation in everyday work. A TET session is an excellent opportunity to use the target software in imaginative ways and socialize with people connected to it.
- (iv) Raising team spirit. The sessions are informal occasions, so getting acquainted with the other participants is possible. In big organizations, the opportunity to meet people from different positions face to face can be quite rare.
- (v) Expanding professional networks and spreading a positive testing attitude. Shared testing experience might ease communication between testers and other project members.
- (vi) Gaining supported hands-on experience with the target system. This could be useful for management and new project members in general.

5.2. Participants. Successful selection of the TET session participants enables the session to reach its goal. A team is most likely to bring sufficient talent and expertise to bear on its task when it has an appropriate number of members with a good mix of skills [27].

Number of Participants: 5–10. According to the earlier group ad hoc testing experiences, participants are hard to get, and studying and reporting results require a lot of effort. Five to ten participants ensure different testing techniques and viewpoints as well as discussion and knowledge sharing. The small size ensures effective communication among all team participants [34], the effort made to arrange a session is not too great, and sessions are manageable. Too large size might lead to serious coordination and motivation losses [27].

Criteria for Invitation: A participant is a team player and has testing experience or domain knowledge.

According to both interviewed group ad hoc testing practitioners, it is beneficial to have participants with different backgrounds. This ensures that the target software is tested from different viewpoints, and communication during sessions is richer. According to Carmel [34], “Of course, diverse teams bring advantages as well: They broaden thinking, they generate more problem solving options, and there is less groupthink.” Familiarity with the target system compensates for a lack of testing experience. This is contradictory to requirements for basic ET practitioners whose testing experience and skills are underlined by testing textbooks (e.g., [8, 25, 35]). During TET sessions, part of facilitator's role is to support ET. Bach noted that, when properly supervised, less skilled testers can find useful results that would not have been found by a script [1]. Also, TET session reports written by the participants are reviewed by the facilitator, who investigates each report before sending it forward. Thus, the quality of the reports is ensured. If the session goal is not to find defects but, for example, to learn the target software, the testing experience is even less important.

Also, each participant's team player skills have to be considered. Stellman and Greene [39] stated “People need to be able to work together, and if they cannot, it does not matter how good their skills are.” The team synergy is the major factor that distinguishes the TET sessions approach from basic ET; therefore, teamwork has to be emphasized. A positive atmosphere supports knowledge sharing and innovativeness.

5.3. Target Software Version and Focus Area. In a TET session, the latest available target software version must be used to get valid results. Before each TET session, the software version is briefly smoke tested to be sure it is testable. Smoke testing target is to assess the build and expose most critical defects [40], and it can be done using the ET approach. If severe regression is revealed, it must be considered whether previous working software should be used instead. The TET facilitator does the smoke testing.

Each participant uses the same version of the tested software, and each device is *configured* exactly the same way. The TET facilitator takes care of the tested devices and their configurations. This ensures that all participants have correct versions of the software, configurations are working, and found defects are not caused by improper settings or old software versions. Also, session time is saved when no extra checks or updates are needed.

Focus area might be a certain target software functionality or quality attribute. If the target system has some area where defects cluster or its test coverage is low, it is a good candidate to be a TET session focus area. Investigating target software defect trends and TCT asset qualities helps to aim the focus area. Also, if the target system is very large, focusing on some specific area might help in getting results.

5.4. Time. The optimal TET session *duration* is, at a minimum, two hours. According to Black [2], the duration of an ET session varies from one to three hours and must include several iterations in order for the practitioner to

learn the test target, which is essential for defect finding. Both group ad hoc testing interviewees thought that a one-hour duration is enough, especially in a hectic organization where people are busy and have several meetings per day. If a session is longer, it might be hard to get people to participate. As a result of previous experiences and theory, the first TET session was planned for two hours to make sure that several testing iterations could be done. When a TET session requires a short setup, discussion, writing findings down, and possible socializing, one hour gets very short. During the next iterations, the duration was reevaluated based on the realized first session's duration, how time was used in the session, and the feedback received from the participants, but the duration remained two hours.

Frequency. It is good to keep at least two development iterations between TET sessions because arranging sessions requires considerable effort, and there should be enough time for earlier findings to be fixed and new features to be made.

Point in Time. It is wise to arrange TET sessions at the beginning or in the middle of development iteration. At the end of the iterations, people are usually busy finishing tasks. TET sessions should not disturb the flow of the development.

5.5. Tools. All tools utilized in TET sessions have to be prepared well. If a team lacks the tools it needs, its performance surely will suffer even if it otherwise would be effective [27].

If *testing tools* like test frameworks, test case generators, static analyzers, coverage analyzers, or reporting tools are utilized, they have to be prepared and checked to ensure that they are working. Also, they have to be familiar to all participants or so easy to digest that an introduction at the beginning of the session is sufficient; an example is the usage of memory consumption tools if memory leakage detection is the TET session's focus area. Also, one idea is to combine tool training and a TET session. This requires some domain knowledge from the trainer but could contribute to participation, learning, and future use of the tool.

If specific *user accounts* are needed, they must be created—if suitable test accounts do not exist—and prepared by the facilitator before the session. Test user names and other account parameters have to fit the context and be correct. Using personal accounts for testing purposes would violate good testing procedures and could cause security risks. This same applies to possibly needed *test data*.

To support the findings, *reporting templates* are essential. Filling templates should not require too much time but give a basic idea of the reported findings such as defects. Templates should not be too complex; otherwise, they slow down and isolate the reporter from the situation. The TET facilitator takes care of the templates.

5.6. Test Techniques. Among researchers, ET is not seen as a test technique but as an approach that can be applied to any technique. Any testing technique can be utilized in an exploratory way (e.g., [7]). In TET sessions, test techniques might be utilized to make testing more efficient or to teach

participants new testing skills. If a specific technique is used, it has to be familiar to all participants or so easy to digest that an introduction at the beginning of the session is sufficient. If the technique is demanding and new for the participants, the session target should be learning the technique.

5.7. Environment. The *physical TET session environment* must be easily reached by all participants, peaceful to enable concentration, and not too formal to encourage communication. Personal mobile phones should be kept silent, and no personal laptops should be brought unless they are needed in ET.

Hardware where the target software is installed, as well as network connections and other needed *test environment* variables, must be checked to ensure that they are working and kept under control during a TET session by the facilitator. It is optimal that the TET participants do not have to bring or prepare any testing-related equipment.

5.8. Introductions and Reporting. Because unguided exploration can be cumbersome and inefficient [29], each TET session starts with an *introduction* to the tested software, the TET session approach, and the goals of the session. The TET domain expert introduces the domain area and makes the basic use case introduction. The facilitator introduces the TET session approach and the session goals. This orientation is essential to give the session a strong skeleton, prepare participants with needed facts, and keep people focused. The facilitator also introduces the tools to be used, such as report templates. Introductions should be informative and short. Educational tools such as projectors can be used, but carefully, sessions should not turn into passive situations in which one talks and others listen.

More specific instructions how or what to explore are not given to ensure the team can think creatively. According to Hackman [27], “When a group members get in the habit of thinking creatively about how they will do their work, interesting and useful ideas can emerge—ideas that did not exist before the group invented them.”

The results of each TET session have to be *reported* to domain management and interested parties. Reporting enables approach followup and future TET session planning. In reporting, it is good to consider what is essential and keep the form simple. If the report reflects the goals for the approach, then it is probably working.

5.9. Risks. Approach's risks and mitigation plans are described in Table 2; the key actor is the facilitator unless stated otherwise. Many risks and mitigation plans for reduction relate to the already discussed approach parameters. The better TET session parameters have taken care; the fewer risks are realized.

6. TET Session Process

The TET session process consists of three equally important phases: preparation, the session, and completion. In the following subsections, defect and enhancement idea reporting are used as session goal examples.

TABLE 2: The approach's risks.

Risk and consequences	Mitigation plan
If a clear TET session <i>goal</i> is missing, results will be minimal, time will be lost, and participants and management will feel the method unsatisfactory. On the other hand, too many or contradictory session goals can deteriorate the results.	Each session has to have a clear, context fitting goal. If there are several goals, they have to be in line with each other, and there should not be too many of them. When planning a new session, the facilitator can consult the TET domain expert and management to select best possible goal(s).
If participants lack <i>testing experience</i> , the target software is in poor condition, and the session goal is to find defects, defects are not found or reports might be too incomplete, and results will be lost.	The target software condition and participant testing skills have to be in balance and taken into account during participant selection.
If participants lack an <i>attitude</i> , only obvious defects will be found, and rest of the time will be spent getting bored. If teamwork skills are bad, both information sharing and atmosphere will suffer.	Participant selection has to be done carefully. The facilitator can talk with participant candidates to determine their motivation and attitude.
With a <i>team</i> that is too <i>homogeneous</i> , knowledge sharing suffers. On the other hand, if a team is too <i>heterogeneous</i> , understanding and coordinating with one another can be challenging.	The aspiration in team composing is to strike the right balance between homogeneity and heterogeneity: members should have a variety of talents and perspectives, yet are similar enough to be able to work with each other [27]. To keep the TET session team in balance, participants should have different backgrounds and knowledge bases. This has to be taken into account during participant selection.
Having too many <i>participants</i> causes confusion, and having too few reduces the benefits of the session.	It is good to keep the number of participants between five and ten to ensure results and avoid fuss. For declining participants, there should be back-up candidates.
If the target <i>software version or configuration</i> is not controlled, the results are not reliable and might be caused, for example, by improper settings. With old versions, time is wasted on already fixed defects.	All participants have to have the same, preferably the latest software versions and configurations in use. Participants should not be asked to bring test devices but must prepare to use them.
If the target software is huge and no specific <i>focus area</i> is given, testing efficiency might suffer. On the other hand, if the focus area is already well tested, using it might cause frustration and poor results.	The sessions should have well-chosen focus areas. Analyzing TCT asset and defect metrics can help in selecting the focus area. The facilitator can consult the TET domain expert and management to select best possible focus area(s).
If the session <i>duration</i> is too long, it decreases the number of participants. Neither it can be too short, which would prevent sufficient test cycles and reacting to findings.	A one-hour duration is the minimum session duration. The duration has to be adjusted in accordance with the session feedback and realized session duration. During this research, two hours was found to be sufficient.
If the <i>frequency</i> of sessions is too high, there is a risk of discovering already detected defects.	Leaving at least two development iterations between sessions ensures that there is enough time for earlier findings to be fixed and new features to be made ready for testing.
If the session is organized during the busiest days of the development cycle, it decreases the number of participants and weakens their motivation.	The target software's development cycle has to be taken into account when organizing sessions. At the beginning or in the middle of the iteration, there is more space for additional exercises.
Time is wasted if testing <i>tools</i> are improper or <i>test accounts or data</i> is missing. Inappropriate test accounts or data can cause security problems.	Tools have to be checked to ensure that they are working before sessions take place. Personal accounts or data are not permitted to be used in TET sessions. Specific test accounts and data must be prepared and used.
Without proper <i>reporting means</i> , the results of the session may vanish. If tools are laborious to use, using them might isolate from the situation.	Reporting session findings must be made easy, for example, by using templates that reflect other reporting means of the target project. It is good to ask the name of the reporter if the report needs to be clarified after the session.
If session <i>findings</i> are not reported further, they vanish.	Completion work is an equally important part of the process and, during that, results are reported further. Enough time must be allocated for this work by the facilitator and domain management to make this possible.

TABLE 2: Continued.

Risk and consequences	Mitigation plan
If testing of the target software requires specific <i>testing tools/techniques</i> and all participants are not familiar with them, time and effort will be wasted.	If the testing tool/technique utilized in the session is demanding, the goal of the session should reflect that, and learning should be the main target of the session.
If the <i>technical environment</i> is unstable, results are harder to gain, and results might be unreliable.	The technical environment has to be checked to ensure that it is working (and fixed, if needed), so that testing can be done without interruption.
If the <i>physical environment</i> is restless or too formal, it will disturb concentration and the atmosphere suffers.	The physical environment has to be checked to ensure that it is convenient, so testing can be done in a proper setting. Before the session, the room has to be prepared, for example, templates made available.
If sessions are not <i>guided</i> and focused, it will cause ad hoc testing, testing beside other activities, and resources to be wasted.	The facilitator leads the sessions and keeps the participants focused. The person responsible for facilitator role has to be able to lead a team.
If the session or target software <i>introductions</i> are too formal and long, the atmosphere will stiffen, and time will be wasted.	When preparing introductions, the facilitator and domain expert have to take the informal nature of the session into account. They can also review each other's introductions.
If the session result <i>reporting</i> is missing or inadequate, the benefits of the method are not verifiable and domain management will not be able to do the proper followup. Also planning new sessions will be more laborious and can lead to poor parameter values.	A summary report of the session results must be delivered to all interested parties, and it must reflect the session goals. The report has to document the parameter values used so that they can be taken into account when planning the next session.
If <i>preparation</i> work, that is, parameter definition, is done inadequately, it may cause ad hoc testing, unsatisfactory results, and dropping the method usage.	The facilitator has to allocate enough time for the preparation work; domain management has to make this possible. When defining the parameters, the facilitator can use domain management and expert support.
If the <i>domain expert</i> does not have firm domain area knowledge, the participants will not get the support they need and may end up reporting already known defects and incomplete features.	The domain expert has to be a person who knows the target area and its current situation and is able to express it clearly. This has to be acknowledged when nominating the responsible for this role.
If the <i>TET domain management</i> does not see the benefits of the approach, utilization might fail completely as resources will not be allocated.	Those seeing potential in the approach have to sell the approach to the domain management. Experiences from the other practitioners may help with this. After the first session has taken place, management has to be properly reported to make sure that they have enough information to assess the approach.
No <i>quantitative results</i> gained.	Careful parameter preparing, especially participant selection, lowers the risk. Possible qualitative results, like learning, compensate the outcome.

6.1. Preparation. This phase requires effort, especially from the facilitator. Parameters have to be adjusted to get optimal results. The better a TET session is prepared, the more smoothly it will go, and the better it will serve its purpose. The preparation phase is described in detail in Table 3 below.

6.2. TET Session. TET session results are gained during the session, so it is the core of the whole process. This phase is described in detail in Table 4 below.

6.3. Completion. Proper completion work ensures that the preparation and session effort are not wasted. All findings have to be investigated thoroughly, reporting must be done promptly, and it is good to update the TET session approach according to the experience and feedback attained during the process. This phase is described in detail in Table 5 below.

6.4. Process Summary. Each TET session process phase has role-specific tasks. Actual results are attained during the session phase, but not without proper preparation or completion. During this research, the TET sessions, excluding the preparation and completion phases, took 54 hours— $(7 + 10 + 10 \text{ participants}) \times 2 \text{ hours}$. If all of the hours used are counted together, one session can easily require weeks of investment, so sessions are not free of cost and should be organized carefully. An overview of the approach is presented in Figure 3.

7. Adopting TET Sessions into Industrial Practice

This section describes how to employ the TET session approach into practice and continue using it in a software

TABLE 3: Preparation actions by performer.

Performer	Action
TET facilitator	Defining the TET session parameters: (i) Setting goal(s): when selecting the goal(s) for the session, factors to be taken into account include, for example, the phase of development and the defect situation. (ii) Scheduling: selecting a good point in time for a TET session and deciding on the session duration. (iii) Taking care of practicalities such as space arrangements. (iv) Selecting possible TET focus areas. (v) Selecting TET participants. (vi) Formulating and sending a session invitation and following up on the responses. It is good to briefly introduce the TET session approach and ET in general in the invitation as well as the session goals and possible focus areas. (vii) If applicable: creating/updating defect and enhancement idea templates. (viii) If applicable: creating and preparing/checking test accounts and needed test data. (ix) If applicable: familiarizing with and checking utilized testing tools. (x) If applicable: familiarizing with utilized test technique. (xi) Smoke testing of the target software version. If the latest target software version suffers from severe regression, the session can be rescheduled or a previous working version can be used. ET can be used as a smoke-testing approach. (xii) Configuring and checking software, hardware, network, and other environment variables to be used in the session. (xiii) Preparing the TET approach introduction.
TET participant candidate	(i) Allocating time for the session/rejecting the invitation to let the TET facilitator decide whether she should invite more people.
TET domain expert	(i) Allocating time for the session/proposing a new time. (ii) Supporting the TET facilitator in arrangements, for example, proposing fruitful focus areas. (iii) Preparing the target software (version) introduction, including basic use cases.
TET domain management	(i) Allocating resources for the TET session execution.

TABLE 4: TET session actions by performer.

Performer	Action
TET facilitator	(i) Preparing the session space to save time and start the session smoothly. (ii) Introducing herself to the team. (iii) Introducing the TET session approach, including ET. (iv) Stating the goals of the session. (v) Introducing the tools to be used, such as report templates. (vi) If applicable: introducing the utilized test tools/techniques. (vii) Supporting ET. (viii) Keeping the session focused. The session atmosphere is the key element for success, so the facilitator must work to keep the session focused and easy going.
TET participant	(i) Introducing herself to the team. (ii) Exploratory testing. (iii) Knowledge sharing. (iv) Defect reporting. (v) Enhancement idea reporting.
TET domain expert	(i) Introducing herself to the team. (ii) Introducing the target system, possible focus areas, and the software's possible restrictions. (iii) Introducing the basic use cases. (iv) Supporting ET by answering (technical) questions. (v) Commenting on defect and enhancement idea candidates.

project. To facilitate understanding, the phases are compared to the action research stages, though they do not include research activities but concentrate on practicalities, and a full cycle has to be employed only during the first round.

When a software project considers using the TET session approach, it should nominate a person responsible for the

approach, the facilitator. The facilitator's work then starts with the current situation and the approach targets *diagnosis*. The facilitator finds out how ET in general and TET sessions in particular are already used in the target project or in the same organization. The approach is easier to employ if the project is already familiar with the approaches, and

TABLE 5: Completion actions by performer.

Performer	Action
TET facilitator	<ul style="list-style-type: none"> (i) Investigating and reviewing defect and enhancement idea reports: checking to determine whether defects are reproducible, whether defects have been reported already, whether defects are not enhancement ideas or vice versa, and whether enhancement ideas have already been planned. Asking for more information from reporters if something is unclear. (ii) Defect reporting to error management tool. (iii) Existing defect commenting if new use cases for them were found. (iv) Enhancement idea reporting to the requirement management tool. (v) Test case creating/updating based on the defects found. Each defect has to have a corresponding test case in the project's testware, which then can be used in defect retesting and in regression testing to prevent similar defect leaking again. (vi) Test case creating based on accepted enhancement ideas. (vii) Adding created test cases to the regression test suite. (viii) Test coverage analysis based on defect distribution. If defects cluster in certain area(s), the test coverage of those areas might need to be improved. The development team must also be informed of clustering since it can be a sign of a bad coding. (ix) TET session summary reporting to participants, TET domain management, and other interested parties, and for approach followup. (x) Feedback and idea gathering from session participants for TET session process improvements. (xi) Approach updating based on collected feedback and the facilitator's own observations. (xii) Defect verification and retesting after corrections.
TET participant	<ul style="list-style-type: none"> (i) Answering questions concerning unclear reports. (ii) Giving TET session feedback to the facilitator.
TET domain expert	<ul style="list-style-type: none"> (i) Giving feedback on proposed enhancement ideas. (ii) Giving TET session feedback to the facilitator.
TET domain management	<ul style="list-style-type: none"> (i) Following up on the results. (ii) Giving TET session feedback to the facilitator. (iii) Adjusting resources, if needed.

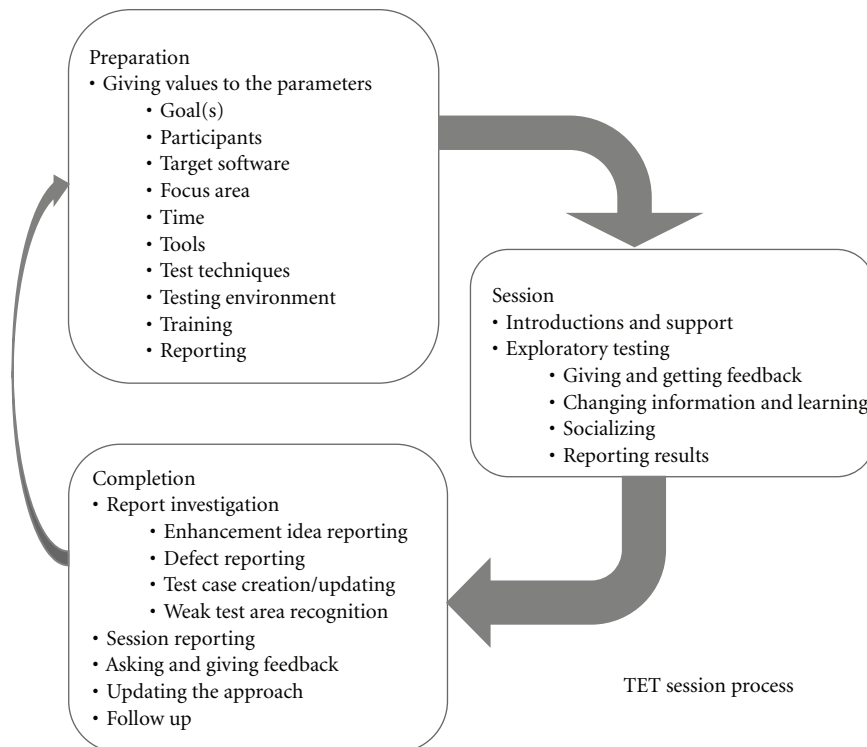


FIGURE 3: Simplified overview of the TET session approach.

current use can offer useful feedback for adjusting the TET approach. The facilitator introduces the TET session approach to project (test) management to make sure that there is necessary knowledge to understand the value of the approach and define targets for utilizing the approach. Having clear goals is essential for any project to succeed [39]. The participants involved in the target definition should present their best knowledge about the target project's testing.

During the *action planning* phase, a rough schedule for the target project's TET sessions is defined: how many sessions there will be (if the approach is not continuous), how often they will be arranged, and when they will take place.

Also, the TET session roles and persons responsible for them are clarified: who is going to take the domain expert responsibilities and who the potential participants are. The last part of the action planning phase is to become familiar with the TET session parameters and process at a detailed level. It is good to adjust the process to the target project's needs and characteristics; the TET facilitator does this.

When the schedule, roles, and process are in place, the facilitator starts *taking action* by giving values to the TET session parameters. It is good to pay special attention to participant selection and clear goal setting. In the TET session, the target system is tested and evaluated by the participants; the facilitator and domain expert support the testing. After the session, the facilitator investigates the findings and reports them further. A summary of the findings is delivered to all interested parties, including the participants and TET domain management, by the facilitator. To keep the approach up to date and suitable for the target project's needs, the facilitator collects feedback.

After the session has taken place and the results have been reported, it is time to *evaluate* the approach: what worked and what needs to be improved. This can be done by assessing the quantitative and qualitative results against the targets defined for the approach. Analysis is done by the facilitator and, based on the results, she updates the approach. The gathered feedback is used as support material. The approach can also be updated during the iteration; *learning* cannot be tightly scheduled but should happen all the time.

After the first iteration has taken place and the TET session approach is employed, the next iterations include only the action taking, evaluating, and learning phases. A summary of the process utilization is described in Figure 4.

8. Assessing the Results against the Target Project's Goals

During each research iteration's analyzing phase, the TET session results were assessed against the goals that the target project had defined for the approach. The target project's management saw the TET session results as satisfying and, here, what participants thought about the new approach is discussed. The quotes below are from the questionnaire answers.

8.1. Gather Feedback and Enhancement Ideas for the Development. Using sessions as a direct feedback channel for the development was a success. The researcher made the observation that the TET domain expert got lots of oral feedback, and questionnaire answers supported the idea of using TET sessions as feedback channels: "I think this was an excellent channel to give feedback. Especially since there was one representative of the development team participating to give immediate answers if the issues faced were caused by their work or the specifications/requirements they were working under."

Two enhancement ideas were approved for the implementation as a result of the TET sessions. The number is low but confirms that TET sessions can be used for enhancement idea gathering. Also, most of the enhancement ideas (four) reported were already taken into account in the product owner backlog. The reports were not a complete waste but supported the needs of those already planned new features.

8.2. Get New Perspectives on the Testing and Complement Other Testing Activities. The TET sessions, naturally, but also ET were a new approach for most of the participants. Learning a new approach was taken positively: "[the] ET session was a new thing for me which was another learning experience, and a good one at that."

According to the feedback, during a TET session, participants hear other viewpoints regarding the target software, and they can use their imagination; and, unlike TCT, they may test anything they like. ET was seen as a complementary approach to TCT: "All testing types have strengths and weaknesses. Combination of testing types gives best result." The ability to find defects not found by scripted test cases was mentioned as a benefit of ET. A participant who was working as a target project software developer commented "It was a good place to actually test the application with time instead of quickly twiddling with one or two features."

According to the research's TET session findings, a large amount of test cases (42) were updated (18) and created (24). Also, it was noticed that one functional area clearly yielded the biggest share of new defects. As a result, all areas' test cases were reviewed to determine whether they were up to date and whether test cases were missing. Several test cases were updated and created as a result of this review.

8.3. Find Leaked Defects. Altogether, 35 new defects, including six criticals, leaked from the other both static and dynamic testing practices were found during the sessions. Each round formed a clear peak in the target project's defect inflow.

In the feedback, the TET sessions were described as effective because lots of new defects were found in a short time. It seemed that seeing other participants filling out reports encouraged the participants to concentrate even more on defect hunting and reporting. The researcher noticed that, during the sessions, the same defects were found several times and that critical defects already found but not corrected kept coming up. When defects were found to have been reported already, participants often tested around them and found new defects or new use cases for the existing

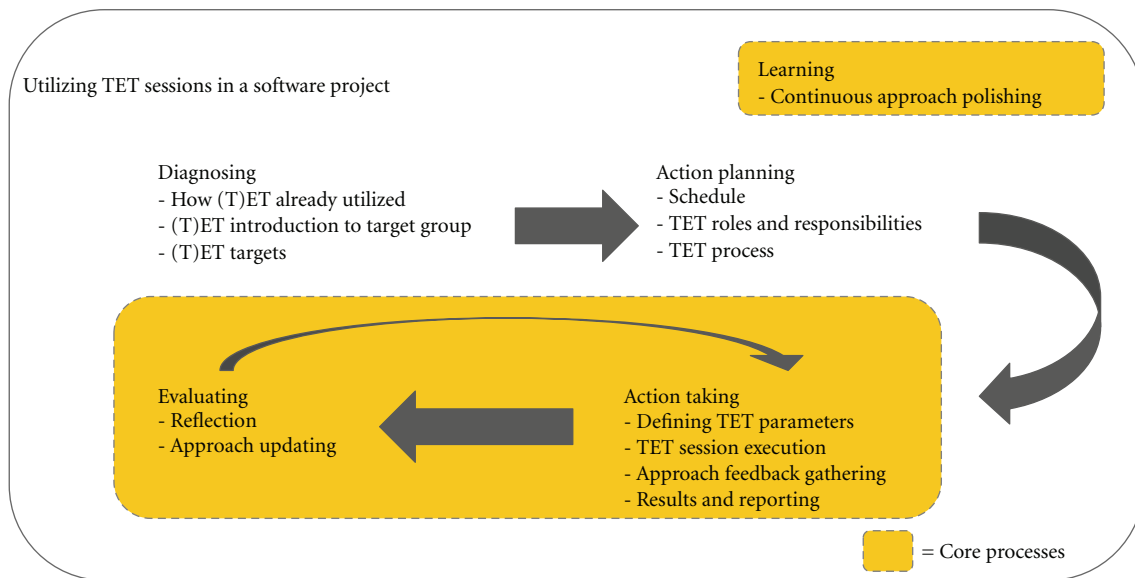


FIGURE 4: How to utilize TET sessions in a software project.

defects. It was concluded that if something is broken in the feature, there might be also other defects waiting to be revealed. Defect clustering is a testing principle [32].

In the questionnaire answers, it was realistically said that, sometimes, participants can find lots of defects and sometimes none. To tackle this problem, “You have to choose wisely whom you are going to invite to session.” The importance of participant selection cannot be highlighted enough.

8.4. Social Effects and Learning. Social effects and learning were taken into account in the researcher’s observations and the questionnaire sent to the participants.

The researcher noticed that the TET domain expert was very busy in hearing feedback, answering technical questions, and explaining why something was done in a certain way. The participants wanted to know more about the tested software. One target software developer even answered that he had learned a bit more about the tested product since he had had no opportunity before to test it thoroughly. New test team members used the sessions as learning opportunities: “As a new test engineer, this session was really good to learn the application and how it works and does not work.”

People were present who had not met before, so new contacts were made. Two participants mentioned that, as a result of the session, they made contacts on the development team. Mixing roles in the TET sessions is beneficial, and the informal atmosphere supports the creation of contacts and conversation. According to the researcher’s observations and the questionnaire answers, the atmosphere was good: “Supporting and very inspired atmosphere.” In the feedback, nobody mentioned raising team spirit but, based on the overall quite positive feedback, it is possible that it occurred,

and, certainly, the sessions provided a nice break from hectic, everyday work.

8.5. The TET Session Approach. The TET session process and the arrangements in general received lots of feedback. This particular piece of feedback came from a person who had participated in previous group ad hoc testing sessions: “Big plus is that there was phones and test accounts ready. Also that Scrum master who knew much about the product was good plus so that it was easy to verify whether the finding was a bug or not.” Other tools, that is, templates, were said to be clear and understandable.

All time-related parameters received feedback. The two-hour duration was commented on as being just enough for one session. On the other hand, one participant would have liked to test even longer, and one wrote that the maximum duration should be 1.5 hours. Also, it was stated that it is good to have a one-month break before arranging the next session. This was taken into account in the session frequency adjustments during the research. The sessions were scheduled at the beginning of the sprints, and one developer participant commented that, at the end of the sprint, he would not have been able to participate.

Support was acknowledged to be sufficient: “Whenever there were unclear points there was enough support to clarify them.” The participation of the domain expert was mentioned many times, and also it seemed that the process approach was seen as positive: “Session was very efficient. Good introduction and preparations had been made well. Devices, sim cards, tested SW and error reporting templates were in place. Also it was good to have scrum master as commentator for tricky questions. Coffee and pulla [bun] inspired to participate into the session.”

Also, ET as a session work method received feedback. The following characteristics were identified as ET benefits: the possibility of finding more random defects because ET utilizes more uncommon use cases. ET emphasizes the freedom to explore, which might lead to defects not found by TCT. ET is testing from the end-user's point of view, and ET improves the testing beyond test case-based testing. A disadvantage of ET was the difficulty of documenting the steps leading to found defects. During ET, participants do lots of things, and it might be difficult to isolate the steps leading to a specific defect. Other recognized disadvantages of ET were the following items: ET is not comprehensive or sufficient as such, it is sometimes hard to find out what to try, and ET requires an interest in the tested product: "It was fun for a time, but to me, it felt rather aimless and boring. You really have to be completely and absolutely thrilled about the product to become a good exploratory tester."

9. Threats to Research Validity

The research session participants were very motivated because the TET session approach was new and, hence, tempting to try. Therefore, the results could have been too positive, and maintaining this enthusiasm requires concern. On the other hand, when participants get more practice with the approach, their skills may also sharpen.

The internal validity, which concerns relations between concepts, can be threatened by incorrect results from the different sources of information. Action research as a qualitative method suggests multifaceted observations and data gathering. The research memos, interviews, observation notes, and questionnaires are typical tools. Interviews and questionnaires were used also in this research as channels when collecting lessons-learned topics. There was no threat posed by the misuse of statistical analysis because only basic numerical calculations were used in the research.

The research did not use any control group for comparing the results gained by utilizing the approach. Effect of utilizing the approach was measured by collecting both quantitative and qualitative results of using the approach. The research was done inside one company, so generalizing especially the results gained with the approach is hard; they depend on the approach goals and context like all testing results do.

From the viewpoint of external validity, that is, to the extent to which it is possible to generalize the findings, we see that the approach is systematically explored here for the first time; therefore, it requires more research. Drawing general conclusions is risky. It cannot be assumed that the results of the study are, as such, ready to be generalized and implemented in other organizations. Practitioners have to define the parameters case by case and, further, during each iteration, they must update the approach. In this research, the session goals were outlined only preliminarily, the majority of the participants had testing experience, and the parameters were kept at a quite general level. Determining the purposes for which the TET session approach can be used and how the parameters, roles, and process can be adjusted

to serve those different purposes is a subject requiring more research.

10. Conclusion

The contribution of this research is in introducing the TET session, an approach for managing *team* ET. A TET session is an uninterrupted, reviewable, chartered, and focused team testing unit. The team tests a specified focus area in the target system and has common goals, and as a common working method: ET. Participants have different backgrounds, but not necessarily testing knowledge, to enrich communication and learning. They do not have to prepare for sessions but must be willing to share their knowledge and communicate with each other. Each team member has a specific role, and a facilitator leads the sessions. The facilitator takes care of the preparation and completing tasks to keep the threshold for participation low and make it possible for people other than testers to participate. Nontester participation is an advantage of the approach, people not doing testing as part of their daily job get hands on experience of the target system, and the target project benefits from their findings that might differ from testers' findings. A domain expert supports the participants technically, shares target software-related knowledge, and takes feedback. Team enables a synergistic gain: when team members interact in ways that help them learn from one another, they increase the total pool of talent available for the work at hand [27].

The TET session parameters that have to be taken care of for each session are the goal, participants, target software version, focus area, time, tools, test techniques, testing environment, introductions, and reporting. Careful parameter handling manages approach-related risks. The roles are the facilitator, domain expert, participant, and domain management. Each role has different responsibilities within the process. The process consists of the preparation, session, and completion phases. All have to be executed carefully to get optimized results. The results depend on the goals; they can be, for example, finding defects, learning the system, or learning a new test technique.

The quantitative and qualitative results of utilizing the approach indicate that it is worth doing. During this research, as a result of three TET sessions, 35 new defects, including six criticals, were recognized, 18 test cases were updated and 24 created, two enhancement ideas were approved to implementation, and one weak testing/coding area was recognized via defect clustering. The results were not mathematically compared to other ET or TCT but reviewed by the target project's management and accepted as satisfying. Qualitative results were also reached: application knowledge and testing skills were improved. New contacts were made, and the sessions offered nice but productive breaks from everyday work. How all these results were achieved is not possible to measure quantitatively so asking for feedback from the participants is an essential part of the TET session approach.

The use of TET sessions was found to support traditional test case-based testing and individual ET, but the benefits are not achieved without investment. Arranging sessions

requires effort; each phase of the process has to be paid attention. ET is creative by nature, and the purpose of the TET session approach is not to stiffen it. The approach focuses sessions and makes participation effortless so that the team can concentrate on session goals. The use of teams distinguishes the approach from basic ET and gives the approach its value. Also, the criteria for TET session participation are different since no preparation or even test knowledge is required of participants; instead, knowledge of the target software and team player skills are emphasized.

To arrange a successful TET session, one must prepare well by, for example, finding suitable participants, fitting the process to the target project's working methods, keeping the session focused, and investigating the findings thoroughly. When this is done, both the target system and the team benefit. During this research, the target system improved due to defect fixes and approved enhancement ideas, and its testing coverage rose through the creation of updated and created test cases. Weak testing/coding area recognition gave important input for both the development and the testing. The team got better by gaining knowledge about the target system and its background, testing, and each other.

The approach is explored here for the first time, so it requires more research. Trying this approach and developing it further would offer an interesting topic for a study. In this study, the approach was utilized in agile Web application development project where ET is suitable [41], so trying the approach in a nonagile project of another type could be worth studying. Also, identifying the varying purposes for which the TET session approach can be used and how the parameters, roles, and process can be adjusted to serve those different purposes is a subject requiring more research. Interesting topics could include how to effectively use testing tools or techniques in TET sessions. TET session testing does not have to be manual, and the approach can be used with any test technique. For example, combining tool or technique training in a TET session could be a fruitful combination and research subject.

Comparing TET session and individual ET session results would be interesting. The team is the central part of the TET session approach, so the importance of teamwork could be investigated further. For example, comparing learning efficiency, determining the proportion of findings that are duplicates of already reported findings, and identifying how different participation criteria—basic ET requires at least some testing experience, but TET session do not necessarily require it—affect the session results could be interesting avenues of research.

References

- [1] J. A. Bach, "Exploratory Testing Explained," 2003, <http://www.satisfice.com/articles/et-article.pdf>.
- [2] R. Black, *Pragmatic Software Testing. Becoming an Effective and Efficient Test Professional*, Wiley, Indianapolis, Ind, USA, 2007.
- [3] G. J. Myers, *The Art of Software Testing*, John Wiley & Sons, New York, NY, USA, 1979.
- [4] J. Itkonen, *Do test cases really matter? An experiment comparing test case based and exploratory testing*, Helsinki University of Technology. Department of Computer Science and Engineering, 2008.
- [5] A. Abran, J. W. Moore, P. Bourque, R. Dupuis, and L. L. Tripp, *SWEBOK. Guide to the software engineering body of knowledge: 2004 version*, Angela Burgess, 2004.
- [6] M. Bolton, "Exploratory testing explained," in *Agile Testing. A Practical Guide for Testers and Agile Teams*, L. Crispin and J. Gregory, Eds., pp. 195–198, Addison-Wesley, Upper Saddle River, NJ, USA, 2009.
- [7] C. Kaner, "A tutorial in exploratory testing," 2008, <http://www.kaner.com/pdfs/QAExploring.pdf>.
- [8] J. Kohl, "Exploratory Testing on Agile Teams," 2005, <http://www.informit.com/articles/article.aspx?p=405514>.
- [9] Ja. Bach, "Where does exploratory testing fit?" 2001, http://www.satisfice.com/articles/where_fits.shtml.
- [10] Ja. Bach, "General functionality and stability test procedure for certified for Microsoft Windows logo," 1999, <http://www.satisfice.com/tools/procedure.pdf>.
- [11] J. Itkonen and K. Rautiainen, "Exploratory testing: a multiple case study," in *International Symposium on Empirical Software Engineering (ISESE '05)*, pp. 84–93, November 2005.
- [12] M. E. Fagan, "Design and code inspections to reduce errors in program development," *IBM Systems Journal*, vol. 15, no. 3, pp. 182–211, 1976.
- [13] E. Yourdon, *Structured walkthroughs*, Yourdon Press, 1978.
- [14] K. E. Wiegers, *Peer Reviews in Software: A Practical Guide*, Addison-Wesley, Boston, Mass, USA, 2002.
- [15] O. Laitenberger, "Cost-effective detection of software defects through perspective-based inspections," *Empirical Software Engineering*, vol. 6, no. 1, pp. 81–84, 2001.
- [16] F. Shull, I. Rus, and V. Basili, "How perspective-based reading can improve requirements inspections," *Computer*, vol. 33, no. 7, pp. 73–79, 2000.
- [17] J. Tuomikoski, *Absorbing software testing into the Scrum method*, University of Oulu. Department of Information Processing Science, 2008.
- [18] J. Hagar, "Technique: exploratory testing and information evaluation," in *Agile Testing. A Practical Guide for Testers and Agile Teams*, L. Crispin and J. Gregory, Eds., pp. 198–200, Addison-Wesley, Upper Saddle River, NJ, USA, 2009.
- [19] G. I. Susman and R. D. Evered, "An assessment of the scientific merits of action research," *Administrative Science Quarterly*, vol. 23, pp. 582–603, 1978.
- [20] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering*, vol. 14, no. 2, pp. 131–164, 2009.
- [21] R. L. Baskerville and A. T. Wood-Harper, "A critical perspective on action research as a method for information systems research," *Journal of Information Technology*, vol. 11, no. 3, pp. 235–246, 1996.
- [22] D. E. Avison, F. Lau, M. D. Myers, and P. A. Nielsen, "Action research," *Communications of the ACM*, vol. 42, no. 1, pp. 94–97, 1999.
- [23] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS Quarterly*, vol. 28, no. 1, pp. 75–105, 2004.
- [24] L. Crispin and J. Gregory, *Agile Testing. A Practical Guide for Testers and Agile Teams*, Addison-Wesley, Upper Saddle River, NJ, USA, 2009.
- [25] Ja. Bach, "Exploratory Testing and the Planning Myth," 2001, <http://www.satisfice.com/articles/et.myth.shtml>.
- [26] B. Boehm and V. R. Basili, "Software defect reduction top 10 list," *Computer*, vol. 34, no. 1, pp. 135–137, 2001.

- [27] J. R. Hackman, "The design of work teams," in *Handbook of Organizational Behavior*, J. Lorsch, Ed., pp. 315–342, Prentice-Hall, Englewood Cliffs, NJ, USA, 1987.
- [28] J. M. Carroll and A. P. Aaronson, "Learning by doing with simulated intelligent help," *Communications of the Association for Computing Machinery*, vol. 31, pp. 1064–1079, 1988.
- [29] J. Rieman, "A field study of exploratory learning strategies," *ACM Transactions on Computer-Human Interaction*, vol. 3, no. 3, pp. 189–218, 1996.
- [30] J. M. Carroll et al., "Exploring a word processor," *Human-Computer Interaction*, vol. 1, no. 3, pp. 283–307, 1985.
- [31] B. Wong and M. Bhatti, "The influence of team relationships on software quality," in *ICSE Workshop on Software Quality (WoSQ '09)*, pp. 1–8, May 2009.
- [32] A. Spillner, T. Linz, and H. Schaefer, *Software Testing Foundations. A Study Guide for the Certified Tester Exam*, Rocky Nook, Santa Barbara, Calif, USA, 2nd edition, 2007.
- [33] J. R. Katzenbach and D. K. Smith, *The Wisdom of Teams: Creating the High Performance Organization*, HarperBusiness, London, UK, 1999.
- [34] E. Carmel, *Global Software Teams. Collaboration Across Borders and Time Zones*, Prentice Hall, Upper Saddle River, NJ, USA, 1999.
- [35] T. DeMarco, *Controlling Software Projects: Management, Measurement, and Estimates*, Yourdon Press, New York, NY, USA, 1982.
- [36] Jo. Bach, "Session-Based Test Management," 2000, <http://www.satisfice.com/articles/sbtm.pdf>.
- [37] J. Lyndsay and N. van Eeden, "Adventures in session-based testing," 2003, <http://www.workroom-productions.com/papers/AiSBTV1.2.pdf>.
- [38] C. Kaner, J. Bach, and B. Pettichord, *Lessons Learned in Software Testing. A Context-Driven Approach*, John Wiley & Sons, New York, NY, USA, 2002.
- [39] A. Stellman and J. Greene, *Beautiful Teams. Inspiring and Cautionary Tales from Veteran Team Leaders*, O'Reilly, Beijing, China, 2009.
- [40] R. S. Pressman, *Software Engineering. A Practitioner's Approach*, McGraw-Hill, Boston, Mass, USA, 7th edition, 2010.
- [41] J. A. Whittaker, *Exploratory Software Testing. Tips, Tricks, Tours, and Techniques to Guide Test Design*, Addison Wesley, Upper Saddle River, NJ, USA, 2009.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

