ORIGINAL ARTICLE

Workshop videos for requirements communication

Samuel A. Fricker[1,4] · Kurt Schneider[2] · Farnaz Fotrousi[1] · Christoph Thuemmler[3]

Received: 19 August 2014 / Accepted: 5 June 2015 / Published online: 22 June 2015

Abstract Shared understanding of requirements between stakeholders and the development team is a critical success factor for requirements engineering. Workshops are an effective means for achieving such shared understanding. Stakeholders and team representatives can meet and discuss what a planned software system should be and how it should support achieving stakeholder goals. However, some important intended recipients of the requirements are often not present in such workshops: the developers. Thus, they cannot benefit from the in-depth understanding of the requirements and of the rationales for these requirements that develops during the workshops. The simple handover of a requirements specification hardly compensates the rich requirements understanding that is needed for the development of an acceptable system. To compensate the lack of presence in a requirements workshop, we propose to record that requirements workshop on video. If workshop participants agree to be recorded, a video is relatively simple to create and can capture much more aspects about requirements and rationales than a specification document. This paper presents the workshop video technique and a phenomenological evaluation of its use for requirements communication from the perspective of software developers. The results show how the technique was appreciated by observers of the video, present positive and negative feedbacks from the observers, and lead to recommendations for implementing the technique in practice.

Keywords Workshop videos · Requirements communication · Video analysis · Developer perception · Phenomenological evaluation

1 Introduction

Whenever a group of stakeholders has requirements for a software or a system product, one of the most fundamental challenges is how to communicate those requirements to developers in an efficient and effective way [1]. If this communication is successful, a shared understanding is created between stakeholders and developers [2]. It is the basis for developing software that meets stakeholder needs.

Depending on the development method, there are different ways of conveying requirements information to developers. Models of software development such as ISO/IEC/IEEE 29148 demand documents to support that flow of information. Alternatively, requirements communication may be seen as a control process and implement feed forward and feedback to improve requirements communication success [3]. Agile approaches embrace such

✉ Samuel A. Fricker
samuel.a.fricker@gmail.com; samuel.fricker@bth.se;
samuel.fricker@fhnw.ch

Kurt Schneider
kurt.schneider@inf.uni-hannover.de

Farnaz Fotrousi
farnaz.fotrousi@bth.se

Christoph Thuemmler
c.thuemmler@napier.ac.uk

[1] Blekinge Institute of Technology, SERL-Sweden, Campus Gräsvik, 371 79 Karlskrona, Sweden

[2] Leibniz Universität Hannover, Welfengarten 1, 30167 Hannover, Germany

[3] Edinburgh Napier University, 10 Colinton Road, Edinburgh EH10 5DT, UK

[4] FHNW, School of Engineering, Klosterzelgstrasee 2, 5210 Windisch, Switzerland

feedback and foster direct communication between stakeholders and developers throughout the whole project [4]. In many companies, however, such collaborative requirements work is constrained to workshops [5].

Workshops are used to engage stakeholders in joint decision making [6, 7]. For stakeholders of a software system, workshops are an interactive forum to learn about their viewpoints and to agree on a shared understanding of requirements for the software system [2]. The direct communication in a workshop includes immediate feedback that avoids delays and indirections. This approach prevents mistakes, identifies and resolves conflicts, and fosters an agreement that is supported by the stakeholders. The rationales and priorities used by different stakeholders for decision making are central for communicating requirements and guiding design [8].

Many situations do not allow seizing the benefits of workshops, however. A requirements workshop is held before many of the developers have been assigned to the project. In public tenders where the requirements specification work is strictly separated from the offering and delivery of a solution [9], developer involvement would even be problematic. In such cases, developers are not able to see and hear stakeholder interaction, nor can they follow discussions and negotiations at first hand when requirements get elicited, clarified, and justified.

This paper proposes video recording of requirements workshops as a technique to communicate rich information about requirements from stakeholders to developers. Video recording has been pioneered for communicating knowledge already for almost 150 years [10]. In the 1870s, series of images were used to document animal and human movements. In the 1890s, the first videos were created to document how humans interact with technology. Over time, such capture of human–technology interaction with video was taken up in workplace studies, human–computer interaction, and computer-supported cooperative work. In requirements engineering, videos of human–computer interaction were used to document system context [11], product vision [12–14], or scenarios [15–17] and used as an input to requirements workshop, to analyze usability [18, 19] or to build specifications [20]. However, no research explored how videos produced as an outcome of requirements engineering are perceived by developers who would use the video as an input for development.

This paper introduces the use of workshop videos for requirements communication and evaluates the technique from the perspective of developers who get the videos as a representation of requirements. To understand what a workshop video means for a developer, we let 18 advanced software engineering students with software development experience evaluate a video of a real requirements workshop. For evaluation, they took the perspective of developers who would implement the requirements discussed in the video. The laboratory evaluation was replicated with the head designer of the system discussed in the workshop video. The results indicate that most of the laboratory results can be translated into real-world projects.

The aims of our static laboratory evaluation are the identification and resolution of problems before a technology is tested in production projects. Thus, we consider it an important step for transferring technology from academia to industry [21]. The results presented in this paper show (1) how the technique is appreciated by developers, (2) factors that affect its quality, and (3) recommendations of how to implement the technique in practice. The obtained insights are essential for calibrating the workshop video technique and for defining guidelines for its effective use to communicate requirements.

The remainder of the paper is structured as follows. Section 2 describes the concept of workshop videos and explains how the technique contributes to solving the requirements communication problem. Section 3 describes the research methodology used for laboratory evaluation of the workshop videos. Section 4 shows the results obtained with the laboratory evaluation, including appreciation, positive and negative aspects, and recommendations for implementation of workshop videos in practice. Section 5 presents the real-world replication of the evaluation and discusses the threats to validity of the presented study. Section 6 discusses the obtained results, including contribution and implications. Section 7 summarizes and concludes.

2 Workshop videos for requirements communication

2.1 Requirements communication

According to Fricker [3], requirements communication is the *process of conveying needs from a given customer to a given supplier who enables the latter to implement a solution that is accepted by the former*. This definition is valid also for a development project, where the customer is represented by a set of stakeholders and the supplier by the development team. Successful requirements communication leads to a shared understanding and agreement between the stakeholders and the development team about what the relevant requirements are [22] and what the meaning of these requirements are in terms of the system that is to be developed [23].

Glinz and Fricker [24] present a variety of practices to build explicit and implicit shared understanding of requirements. Approaches to build explicit shared understanding include domain modeling [25], problem and solution modeling [26], mind maps, glossaries, and ontologies [27]. Each approach leads to an explicit documentation of requirements in a specification. Many of these specifications are also used as knowledge representations that support other software engineering tasks [28]. As an alternative to the explicit representation of shared understanding, joint design of a system, for example with joint prototyping [29], workshops [5], and referencing of known systems allow establishing an implicit shared understanding of requirements. Commonly, the results of these activities are documented in a report about the activity without making the requirements explicit however. It depends on the members of the development project and their tasks to determine the most appropriate format for building shared understanding.

To minimize requirements communication problems, modern development processes advocate real-time interaction between stakeholders and the development team [4]. Workshops are used frequently for that purpose [30]. Requirements workshops generate requirements of high quality, build trust, and enhance the communication between participating stakeholders and members of the development team. With careful preparation and the guidance of a neutral facilitator, a workshop becomes an effective means for discovering requirements, assigning priorities, and establishing agreement on the requirements.

According to Voinov [6], workshops are used for engaging stakeholders in joint decision making. Workshops are a forum to learn and build a shared understanding that enables innovation. Workshops help those who will be bearing the consequences of the decisions to translate their individual viewpoints into a common language and a coherent whole. As a result, better decisions are implemented with less conflict and more success.

Voinov, Persson, and Stirna [6, 7, 31] suggest that workshop participation be facilitated by method experts who do modeling and use scenario walk-throughs, prototypes, and simulations to explore the stakeholders' viewpoints, support learning, and document decisions. The method experts ensure reasonable use of the workshop method and quality of the outcome by building trust, motivating the participants, moderating the group process, and improving when necessary. A sufficient number of participants is actively involved to ensure coverage of the problem domain knowledge and authority to address the problem at hand.

The workshop is started by defining an agenda and agreeing on the rules to prioritize and select requirements. Requirements are then discovered and explored. Scenarios help structuring the discussions for discovering requirements [32, 33]. Prototypes [34, 35], role-play [36], and workplace immersion [24, 34] can be used to generate requirements recognition cues. Models are used to create an integrated and agreed description of the different aspects of the system and discussion [6, 7]. Simulations may further support the workshop by providing means for interactive experimentation and facilitate learning [37]. When concluding the workshop, the generated requirements are tested for correctness and completeness by walking through a summary of the created work results.

Once understood and agreed, the requirements need to be propagated through the project, so that all project members receive the inputs they need. Stapel et al. model the flow of requirements and information in software projects [38, 39]. They emphasize the need to take both document-based and direct communication into account [40]. Kwan et al. [41] follow individual requirements on their way through the project. For effective support of the software engineering tasks at hand, it is important to choose requirements representations and propagation techniques that are adequate for the respective situation.

Unfortunately, direct communication is not feasible in many situations because requirements workshops are held before the relevant developers have been assigned to the project. Some organizations engineer requirements before they launch projects [42], many projects work on requirements before they start significant implementation [43, 44], and staff changes as the software ages and evolves [45]. Developer involvement in requirements workshops would be particularly problematic in public tenders [9], where requirements specification is strictly separated from the offering and delivery of a solution. Thus, developers are not able to see and hear stakeholder interaction, nor can they follow discussions and negotiations first hand, when requirements get elicited, clarified, and justified.

In these situations, projects tend to communicate requirements by handing-off written specifications as suggested by standards such as ISO/IEC/IEEE 29148:2011. Fricker evaluated the impact of such hand-off on requirements understanding and showed that the practice was problematic however [46]. According to the obtained results, the hand-off did not lead to good-enough requirements understanding. The architect and developer of the software solution did not understand the impact of requirements on the design well enough and did not have enough information about the usage of the solution to evaluate the appropriateness of tentative designs.

2.2 Videos to enrich indirect communication

Videos have been used as a rich source of data for the purpose of documentation and research in requirements engineering [11–20] and wide range of areas that involve research how humans interact with technology. The textbook on "Video in Qualitative Research" by Heath et al. [10] provides an overview of the potentials and pitfalls of capturing natural behavior in real-life scenes. There are good recommendations and checklists for preparing video recording. Numerous problems and threats to validity are associated with access to the scenery, ethics of video recording, and impact of a camera on subject behavior.

Videos are considered to be effective for addressing the problem of understanding in situations where direct communication is not possible. Carter and Karatsolis [47] advocated video and other means of rich documentation: "We believe that when used properly, electronic white boards and video cameras can capture the richness of the process far more effectively than notes and recollections. The challenge is to engage people with the right tools, skills, and talents to establish the context and to post-process the results properly. This suggests that research into a different set of tools aimed at capturing requirements and design activities, analyzing these records, and then producing effective clips might be a valuable investment." Zachos [15] also considers rich media a rich source for development. According to Brill et al. [12], developers appreciate videos because they are rich and concrete in comparison with text that is perceived to be more precise but also more abstract.

However, videos may have potentially unintended impacts. For example, individuals may inadvertently say something they would like to erase later. Responsible use of video should be based on identification, reflection, and deliberation of such risks in a dialogue with the affected stakeholders [48]. The goals of this dialogue were to define shared values and rules of how the videos are to be created, processed, and used. Tools to implement responsible use of videos include informed consent for documenting agreed rules [10], video processing for anonymization [49], and erasure of video recordings to allow participants to be forgotten [50].

The use of video has a long history in software engineering. Feeney reported that the graphics, motion, and spoken information provided in a video allow programmers to learn easier than written documentation [51]. In addition, the production of a video was faster than the writing of documentation. DeMarco and Geertgens [52] reported consistent results from using low-cost VHS video recording for program documentation in 1990, with the added benefit of captured rationales: "An additional benefit was that the videos gave some insight about the personality and thought processes of one of the principal designers."

Videos were also used for communicating a product vision and getting feedback on the product that did not yet exist. A well-known example is Apple's 1987 vision video of a personal assistant on a laptop, the Knowledge Navigator.[1] What may look straightforward today was a visionary illustration of features and interactions many of which have been implemented in the meantime. Vision videos illustrate and demonstrate concepts for elicitation and validation, much like a prototype. However, producing the video does not require implementing a single line of code. That helps in eliciting feedback and requirements *early*, by discussing the vision.

Creighton extended vision videos with traceability to requirements [20]. In collaboration with Siemens, product visions were illustrated as high-end marketing videos where users interacted with system components. UML diagrams were overlaid with the videos. This UML video overlay gave the possibility to trace video and requirements and allowed the scenes, actors, components, and actions to be connected to later development activities. However, production and analysis of this type of videos required extensive preparation, a precise storyboard, and sophisticated post-processing.

Videos have also been used to support automated GUI testing and software documentation [53]. They were used to describe the visual aspects of an interface and for providing evidence whether the solution meets its requirements.

2.3 Workshop videos

So far, videos have not been adopted widely in requirements engineering. We believe that the difficulty of creating a high-quality video and the cost of the equipment needed to produce and edit the videos were hindering the adoption in earlier years. In the meantime, these circumstances have changed. Cameras and tools for handling videos have become much more accessible, easier to handle, and even less costly: New technologies such as the widely spread smart phones make video recording available to almost everyone. Thus, video is now a readily available option for capturing and communicating requirements [16].

To make the use of videos practicable, we suggest lightweight and low-effort use of the video technology. Videos can easily be created to document the process of discussing and clarifying requirements as it occurs in a requirements workshop. We propose to use video recording of a requirements workshop to convey the shared understanding that is developed during the workshop. The video is used to communicate the requirements discussed in the workshop to developers who are expected to build on these requirements, but were not able to participate. The recording of a workshop will allow the video recipients to benefit from understanding the rationales that the real stakeholders used for agreeing on the requirements. In comparison with a written report of the workshop, the video recording will provide documentation that is created with low effort and that provides information nearly as rich and trustworthy as the actual participation in the workshop.

[1] http://www.youtube.com/watch?v=JIE8xk6Rl1w.

A critical precondition for the technique is that the use of the workshop video is explained to all parties who are expected to appear in the video, and that consent is obtained from them that they may be recorded. The consent shall describe the processing and use of the video and the individual's right for video erasure. Such informed consent is not only good practice, but also relevant to prevent litigation and other problems with its later use in the requirements engineering process.

To make requirements recordable, we use the exploration of scenarios as a central element of the requirements workshop. Scenarios are helpful in concretizing requirements that otherwise would be vague and abstract [54]. The exploration of scenarios can then be complemented by supporting techniques. For example, ART-SCENE couples scenario walk-throughs with rich-media storyboards [15]. These provide cues for recognizing and discovering new requirements. Role-playing in which multiple people play various roles may be employed to walk-through or experience system. Role-play helps the participants to develop an in-depth understanding of system requirements and their implications, thus increases the quality of the requirements [30]. Prototypes or implementation proposals may be prepared to facilitate walk-throughs of system requirements and role-plays [23, 36]. Alternatively, a prototype or system design may be created during the workshop as a joint design effort [29]. A prototype proposed by the development team and approved by stakeholders is an artifact that represents a shared understanding of how a system under construction shall look.

For the workshop video technique to be practicable in most situations, a simple and low-effort solution is proposed: Preparation effort is kept to a minimum, and video is recorded on the side, not causing interruptions. Storyboards, prototypes, and implementation proposals are not required, but may be used as optional success enhancers. Participants act freely in the workshop, and new insights are welcome to occur during the workshop. Such openness and flexibility are important because requirements workshops are typically held early in a project when scenarios are not fully settled yet and must be consolidated between stakeholders. Thus, scenes are not planned or enacted, but they emerge from "live interaction" of stakeholders. Such a requirements workshop offers a rich and multi-dimensional opportunity for visualizing, explaining, and discussing requirements. One stakeholder can react to others, and the interaction will not only convey scenarios and requirements, but the discussion will also show preferences and allow a glimpse of the personality of the participants.

The video recording of such workshops will preserve some of the advantages of workshop participation: Information is being conveyed not only with text, but also by recording the behavior, mimics, and interactions of different stakeholders. The video of the requirements workshop will give insight into the personality of stakeholders and how they present and prioritize requirements. In comparison with a requirements specification, the information conveyed by the video will be richer and allow the developer to develop empathy for the stakeholders who will judge acceptance of the system.

Due to the economic constraints in many software projects and the capabilities of regular software engineers, we call for only the most basic video skills and equipment. We thus go beyond pre-planned and very expensive settings for creating workshop videos and take a radical position with respect to the choice of tools, skills, and talents required to record and edit videos. Video recording should be feasible and affordable for any software project. No advanced tools, skills, or talents should be required. No scripting or preparation of the recording should be necessary. A good amateur camera, or even a high-end smart phone, should be sufficient for recording. Post-processing should be very limited in terms of time and effort. With these constraints for a low-cost video recording approach in mind, we see great potential in videos to become a natural part of capturing the requirements engineering process and its results.

2.4 Requirements communication with workshop videos

When used for requirements communication, the video acts as a passive observer for the purpose of documentation and replay. The observed workshop contributes with elicitation, discussion, and negotiation of requirements. To give developers a chance to witness the surfacing rationale, emotions, and interaction, scenarios cutting across several stakeholders will be interesting to see. It is, thus, explicitly intended to capture stakeholder interaction in addition to document resulting requirements as the interaction may contain valuable hints on priorities and rationale.

The entire approach was established as a new means for communicating requirements to developers under the above-mentioned circumstances. Videos preserve *requirements* raised during the workshop and also some of the stakeholder *interaction*, for a later time when developers will be selected and ready to learn more about requirements. An essential feature and benefit of our approach is its capability for supporting asynchronous requirements communication. The developers do not need to participate in the workshop and still benefit from it.

Figure 1 shows the conceptual model behind the workshop video approach as a FLOW information flow model [55, 40]: Requirements are discussed in a workshop that is held for the purpose of discussing, negotiating, and validating requirements from different stakeholder perspectives. This activity is recorded on video to allow developers

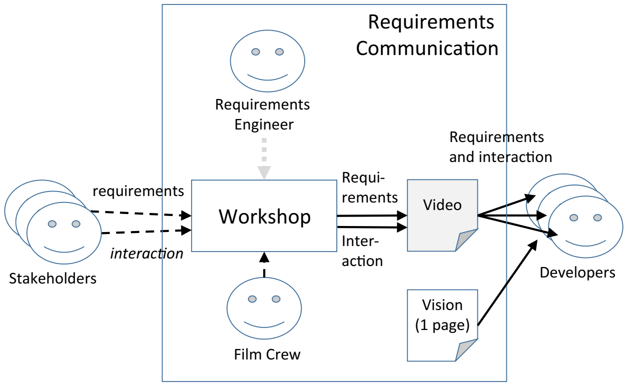

Fig. 1 Information flow diagram of the video approach

to view it and thus obtain requirements discussed—and the dynamics of that discussion—without further indirection. It can even be highly instructive to see where stakeholders are not clear about a use case or have not decided requirements yet.

Initially, stakeholders have their requirements in mind. Stakeholders are encouraged and supported to input their requirements into the Workshop. The video-recorded workshop is supported by a film crew and controlled by a requirements engineer who acts as a workshop moderator. Requirements engineering offers practices for elicitation, interpretation, negotiation, and validation that need to be observed throughout the workshop. For that purpose, the requirements engineer is the one who should lead and moderate the workshop. He or she provides experience in moderation and RE, denoted by the gray arrow to the top of the workshop activity box. Most requirements workshops will follow this pattern, even without a video. Our approach continues beyond that point.

Figure 1 shows a short one-page vision document that was prepared beforehand in the situation presented in this paper and the video as a tangible outcome of the workshop. The video captures both the requirements mentioned verbally and the interaction observed. The information in the video can be retrieved repeatedly, and it can be spread easily, for example by copying and sending it to developers. The film crew is not needed for viewing and spreading the video and the information it contains. Therefore, the video is qualified for storing information over time and for making it available to many developers later (documentation). For that reason, the video is shown as a document symbol.

2.5 Embedding videos in the development process

There are several options to embed requirements workshop videos in the development process. This issue obviously depends on the development model (e.g., plan-driven, agile, lean, prototype-based) and the documents used in that process. Figure 1 shows the information flow actually occurring in our study. In other projects, the process and information flows can be continued in one of several ways, for example

Once developers receive the vision document and the video, they create a specification and continue working with it.

There is some kind of specification document developed in parallel with recording the video(s). One or more videos are used to complement the specified requirements and provide rich information that can be adopted in the specification during an iterative process. This will establish an information feedback loop in Fig. 1.

Depending on the intended process, requirements may be written as a traditional specification, or in the form of epics, story cards, etc.

In Fig. 1, none of these options is demanded or precluded. Most likely, any requirements workshop will produce some documents and other deliverables as a result, including minutes. We expect the need to complement the video with a few other documents and integrate the technique into a bigger-scale process of requirements communication. However, in this paper, we focus on the pure video-only model displayed in Fig. 1 as a prerequisite for embedding workshop videos into various processes later.

3 Evaluation

For any new solution, a clear explanation needs to be provided together with a solid demonstration that the solution is sound [56]. Sound arguments are needed to show that the solution effectively solves the problem it is intended for and that it is a significant improvement over state of art. These requirements apply to the workshop video technique that we here proposed for requirements communication. However, since developer acceptance is important in the use of video workshops for requirements communication and such acceptance can hardly be predicted by argumentation alone, we have decided to go further and provide an early empirical evaluation of the technique.

The aim of the evaluation was to understand the usefulness and acceptance of workshop videos as a technique for communicating requirements from the perspective of a developer who receives the requirements. To achieve this aim, we let potential developers be observers of the workshop video by watching it and reporting how they perceived it for use in a requirements communication context.

We designed the study by asking the following research questions:

RQ1: *How useful is a workshop video from a developer's perspective?*
RQ1.1: *Can requirements be understood with the workshop video?*
RQ1.2: *Is the workshop video perceived useful for requirements communication?*
RQ1.3: *What are the positive, respectively negative aspects of using the workshop video?*
RQ2: *How does a developer judge the quality of a workshop video?*
RQ2.1: *What events in the workshop video are disturbing, respectively helpful?*
RQ2.2: *What is the perceived satisfaction with the workshop video?*
RQ2.3: *What factors affect the usefulness of the workshop video?*
RQ3: *How should requirements be communicated to developers with workshop videos?*
RQ3.1: *Should workshop videos be used?*
RQ3.2: *How should workshop videos be used?*
RQ3.3: *How should workshop videos be produced?*

The answers to the research questions allowed us to understand whether workshop videos should be further explored for requirements communication and how the technique should be tailored for use in a real-world practical context. RQ1 covers the perspective of the developer who receives the workshop video. RQ2 covers quality control of workshop video production. RQ3 identifies recommendations for implementation of the practice.

3.1 Video and observer selection

A video of a requirements workshop from a real software project was evaluated in the presented study. The project aimed at developing a supply chain management system for pharmaceutical drugs. Eight workshop participants explored and agreed on the requirements for the system. The participants were the requirements engineer responsible for requirements specification, the architect responsible for system design, the head of a pharmacy chain that invested in the project, a pharmacist, a lawyer and patient representative, a medical device expert, a selected supplier of barcode readers, and the country head of a barcode standardization organization. Prior to the workshop, a vision statement was distributed, and the concept of operation for managing the reverse supply chain drafted. During the workshop, the participants enacted the drug supply process with real drug packages, barcode readers, and smart phones as mock-ups for exploring and agreeing the system requirements.

The selection of the workshop video is a combination of representative and critical-case sampling [57]. The video featured a requirements workshop with a successful outcome, but with moderation challenges that are encountered in many real-world situations. Overall, the workshop was productive and produced the requirements needed to implement the software discussed in the workshop. However, some of the scenes in the video showed situations that required the intervention of the moderating requirements engineer: a key stakeholder arrived late, one stakeholder occasionally dominated the dialogue, in some situations stakeholders were talking in parallel, and some requirements were discovered that were not anticipated. The presence of these problems allowed us to test whether developers are sensitive to moderation challenges in a workshop video and to generalize statements about video usefulness also to more ideal videos.

The workshop video was recorded by a single person with a handheld camera that included an image stabilization function. To capture the details of devices and artifacts used during the workshops, it was the intention to convey an impression of the entire group of stakeholders, with a focus and zoom on those who discussed at any moment. The entire requirements exploration phase of the workshop was covered. Workshop participant agreement was obtained for the video recording. The camera or film crew was not an active participant in its own right. They were not talking to stakeholders and did not intervene with their interaction. Participants were not instructed previously to show things to the camera explicitly, or act in any way they would not have acted anyway during the workshop. Altogether, the option of *communicating requirements asynchronously* via *video* was provided on the side "as a by-product" [58], and not as an explicit or invasive intervention.

The recorded video covered the workshop in an unedited fashion. We did not remove any information that an editor might be inclined to remove. Avoidance of video editing added the benefit that the evaluators could give feedback on the relevance of video contents without the investigators taking assumptions about whether and which parts of the video are those to be focused on for requirements communication.

We used 18 software engineering students who were at least in their third year of study or higher and who had development experience as observers for evaluating the use of workshop videos. Such "laboratory evaluation" prior to deployment of a technique to real-world practice is common in software engineering research and many other domains. The laboratory evaluation ensures that only well-understood techniques are deployed to practice that have potential for impact and that do not cause harm or annoyance [59]. Research about transferability of research that uses students as proxy–observers for practitioners did not find important differences between students and practitioners if the students have good knowledge of software engineering practices and if their performance is comparable to that of professional developers [60]. In our case, the students were advanced in their studies and had a good understanding of software engineering in general and of requirements engineering in particular. All students had software project experience and had developed software prior to being involved in this study. The participation in software projects with industry gave the students experiences similar to those of many practicing software engineers. The project experience thus increased the relevance of the students' judgments for real-world practice contexts.

Observers were incented for participation in the study by getting access to a real-world case of requirements workshop and by getting study credits. To avoid coercion of any participant, each had the opportunity to opt out of the study and do an alternative exercise of comparable effort. No observer selected the alternative exercise or opted out, however. For receiving the study credits, the video tagging needed to be complete enough, the video tagging rationales rich enough, and the answers in the questionnaire rich enough. One observer was excluded from the study based on these criteria because only one video tag was received.

To understand the transferability of the results to real-world practice, we administered the same research process to the head designer of the supply chain management solution. The results of that validation are shown in Sect. 5.1.

3.2 Data collection and analysis

During data collection, each observer worked individually. The physical location where the video was watched was not constrained. Each observer was given a formal protocol to follow by the observer and two supporting documents.

The protocol included an instruction sheet that stated the observer's goal of *evaluating real-world requirements engineering practice by assessing a requirements workshop from the perspective of a developer*. The sheet then described the tasks to be followed for step-by-step learning about the software solution, experiencing the requirements workshop video, and sharing information about the video experience. The sequence of tasks was as follows:

The investigator informs observers about the research process. The observer gives consent to participate in the study.

The investigator shares the vision of the supply chain management system, asks the observer to take the perspective of a developer for the system, and gives access to the workshop video.

The observer watches the video and tags the video with markers that were annotated with an *interesting*, *boring*, or *comment* label, with a time stamp, and with feedback that gave the observer's rationale.

The observer judges the video, the potential use of the video for development, and recommendations for improving the video by answering a questionnaire.

The investigator reviews the quality of the video annotations and of the answers to the questionnaire and provides feedback to the observer.

The supporting documents introduced the observer into the supply chain management context and, with a 2-sentence problem and solution position statement, to the vision of the supply chain management solution. Other parts of the supporting documentation were the questionnaire described in the appendix of this paper and an explanation of the criteria for when the observer's feedback would be considered good. The criteria were the completeness of the video tagging, the richness of the video tagging rationales, and the richness of the answers to the questionnaires.

The generated annotation data were collected through the video server. The answers to the questionnaire were collected by letting the observers upload the answers into a document database.

To answer RQ1.1 and RQ2.1, the observers tagged scenes they thought to be interesting, boring, or otherwise worth to be commented with markers. Each marker was annotated with a rationale for why the marker was set. They could pause the video and provide short descriptions right away, or they could just attach a marker and make the annotations after watching the entire video. The method we employed is a written variant of the "thinking-aloud" protocol, where the observer immediately responds to a probe [61]. Verbalization that is concurrent with the experience allows the observer to share his or her thinking about the unfolding experience.

The obtained markers and annotations were used to profile the workshop video from the perspective of the receiving developers. The intention of the profile was to give a rich overview of the aspects of the evolving workshop video that generated a reaction of the developer. In an iterative process, the markers were classified into categories that were identified by the researchers from the annotations in a bottom–up fashion. The categories represented the aspects of relevance for observers and were not preconceived by the investigators.

The independent tagging by independent observers who were free to set markers at any time in the video implied that comparable markers had minor time differences. To take this blur on the time axis into account, we used a Gauss kernel as a low-pass filter to aggregate and smooth feedback counts over time [62]. The convolution of the kernel with the signal given by a marker represents the feedback of a Gaussian bell curve on that signal. It is a curve centered on the video time of that marker and spreads to earlier and later times. To even out the influence of observers who provided more markers than other observers, we normalized the marker signals of each individual observer by dividing the respective signal strength by the number of markers set by that observer. The sum of the convoluted bell curves of all normalized feedbacks results in Figs. 2, 3, and 6. The summed convolutions allow adjacent feedbacks to contribute into a common peak. Several feedbacks given around the same time will overlap and contribute to that peak. Peaks exceeding the upper quartile are considered relevant for closer investigation.

To answer RQ1.3, RQ2.3, RQ3.2, and RQ3.3, each observer was asked to reflect about the workshop video he or she had just watched and then answer the questionnaire shown in the appendix of this paper from the perspective of being a potential developer. The appendix describes the detailed mapping of questionnaire and RQ.

To answer RQ1.2, RQ2.2, and RQ3.1, descriptive statistics were used, and the results correlated with the qualitative data that were provided to justify the judgments. The stated reasons for why subjects gave positive and negative judgments, respectively, are a rich source for building models for explaining when and why workshop videos may be effective in requirements communication. The results are thus an important step toward informing large-scale research that aims at validating these models through appropriate hypotheses with statistical methods [63].

Filling-out the questionnaire ensued the tagging of the workshop video and ensured a fresh impression of the video when the observer verbalized his or her opinion about the video-watching experience. Such retrospective probing allows understanding an observer's general, rather than the video episode-specific, interpretation of the workshop video as a means for requirements communication. Thus, it complemented the "think-aloud" protocol with an opportunity for the observer to synthesize all available information in reaction to the questions that were posed in the questionnaire [61].

We performed conventional content analysis [64] of the answers to the questionnaire. A tree structure of categories has been developed and used to structure identified themes for perceived strengths and weaknesses of workshop videos and for recommendations on how to use and improve them. The tree structure is reflected in the results section with the subsection and headers of the tables contained in the subsections. The leaves of the tree correspond to the table entries, which are connected to quotes from the positive and negative answers given by observers. The results give a rich overview of the advantages and disadvantages of the workshop video technique and how to implement it and allowed us to conjecture about the models and theories that might explain the obtained results.

4 Results

4.1 Data collected about the developers' perception of the workshop video

To report about their perception of the workshop video, 18 observers set a total of 451 markers in the video. The video had 48:55 min run time. Figure 2 shows selected frames. The number of feedbacks per observer ranged from 6 to 54 (lower quartile 16, median 25, higher quartile 35). One feedback was excluded due to problems of recording the markers of that observer.

When clustering the markers, the categories of developer feedback emerged: *requirements*, *uncertainties in relation to requirements understanding*, *problems of the requirements workshop*, and *moderation of the workshop by the requirements engineer*. Some markers had multiple classifications when multiple topics were stated in the rationale. No marker was identified that would not fit into any of the categories.

4.2 Usefulness of the workshop video

4.2.1 Requirements understanding

We wanted to know whether the video communicated requirements successfully. According to the markers, the workshop produced many requirements, but had phases

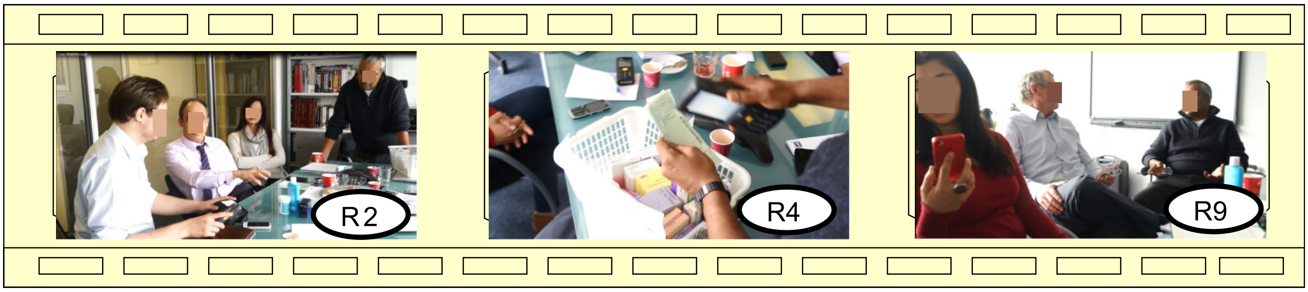

Fig. 2 Scenes from the video related to selected peak of developer feedback (labels refer to events in Fig. 3)

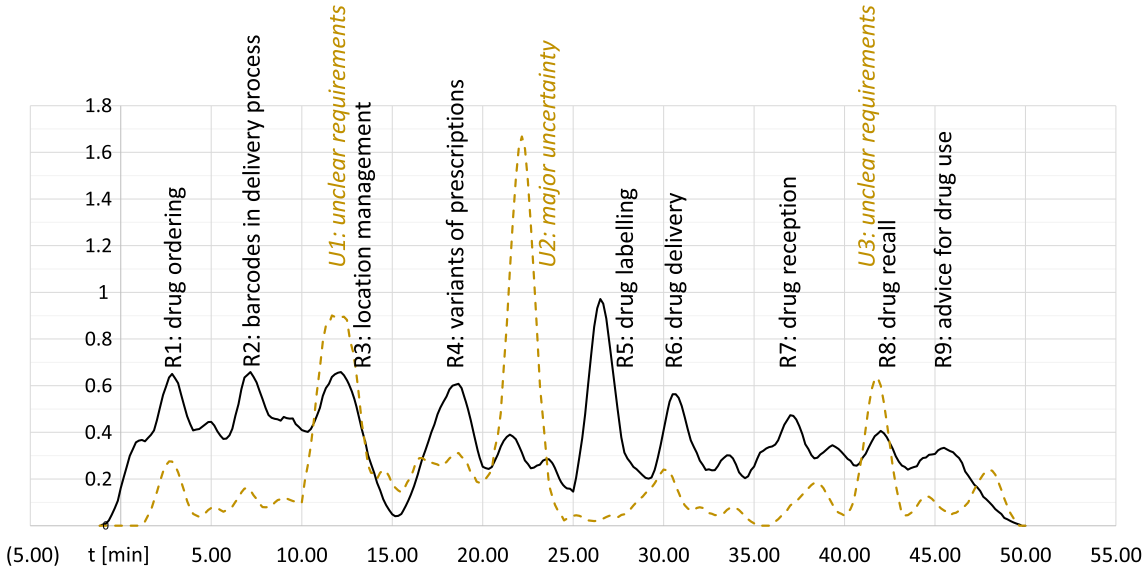

Fig. 3 Requirements communication: requirement topics R1–R9 (solid black) and uncertainties U1–U3 (dashed orange)

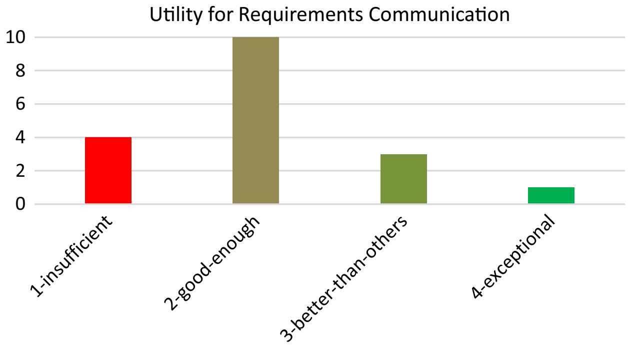

Fig. 4 Votes about usefulness of workshop videos for requirements communication

where requirements were not understood. Figure 3 shows the aggregated feedback about requirements. Remarkable peaks are labeled with *R* if they denote requirements topics identified by the study participants. Nine such topics were identified.

R1: Discussion of drug ordering process.

R2: Discussion of the delivery process to be supported and the types of barcodes to be used in that process. Several people sent feedbacks about this being relevant for requirements, thus creating the peak in the convolved curve.

R3: Requirements about location management.

R4: Requirements about different types of prescriptions and how to handle them.

R5: Requirements about drug labeling in preparation of the dispensing of the drug to the patient.

R6: Requirements about the delivery of the drug for the patient.

R7: Requirements about the reception of the drug by the patient.

R8: Requirements for the drug recall scenario.

R9: Requirements about advice to the patient for drug use.

Peaks labeled with *U* to denote uncertainties about correct requirements understanding. Three uncertainties were identified, and several patterns could be identified:

Both curves run in parallel over extended periods of time. As the tags revealed, a perceived requirement often caused follow-up questions.

There were two peaks of unclear requirements *(U1 and U3)* that were accompanied by peaks of requirements. These peaks seem to be a stronger version of the above-mentioned phenomenon: here, requirements raised *severe* doubts or follow-up questions.

In one case *(U2)*, uncertainties were not directly linked to perceived requirements. Communication of requirements failed at this point.

Shortly after *U2*, there is a peak in requirements reported *(R5)*, and almost no uncertainties reported in the feedback.

When requirements are effectively communicated, follow-up questions may indicate a deep involvement of observers. At the same time, these follow-up questions point to a limitation of one-way communication: follow-up questions cannot be answered. The inability of the video observers to communicate with the workshop participants implies that follow-up questions and uncertainties remain.

The spontaneous feedback was useful to assess the short-term reactions to and acceptance of the requirements workshop video. A prerequisite to communicating requirements with a workshop video is the ability of the video observers to recognize when requirements are being described. Our analysis of the markers showed that observers were able to identify requirements at a high rate throughout most of the video. The analysis showed also that the workshop video was productive despite some problems that the observer feedback made evident.

4.2.2 *Perceived usefulness of the workshop video*

The majority of the observers perceived workshop videos to be useful for communicating requirements. Figure 4 shows the distribution of answers obtained from asking observers "how do you judge the use of video recording for requirements communication?" The large majority judged the workshop video to be good enough according to Regnell's benefit scale [65]. A few judged it to be better than other techniques or exceptional. Four observers judged the video to be *insufficient*.

The *exceptional* rating was justified by the ability of developers to observe the stakeholders and the possibility of watching the video repeatedly. The *better-than-others* ratings were justified by similar arguments, like the video's ability to capture the "why" of the requirements better than other formats, and the video's efficiency in communicating requirements. Some observers desired complementary documentation. The *good-enough* ratings were justified by the same positive arguments. However, concerns were raised about the difficulty of creating a good workshop video and about the limitations of the technique for capturing all relevant data, for resolving a developer's questions, and for storing requirements in a format that is easy to use by developers. The *insufficient* ratings were justified by the perceived lack of structure in the video and lack of confidence of the observer to have the requirements understood sufficiently well. Here, workshop videos were seen as a complement to a requirements document, not as a replacement.

In addition to the usefulness rating, we wanted to know whether the workshop video would enable developers to implement the discussed solution. Many participants said it did. Figure 5 shows the distribution of the opinion scores that were obtained from asking observers "How capable do you feel to be able to implement the solution discussed in the video?" Most observers judged their personal ability to implement the solution to be *good* after having watched the video. The median was between *fair* and *good*. None of the extreme values *bad* or *excellent* were chosen.

The *good* ratings were justified with the availability of enough information to start designing and implementing a solution and with the presence of the architect in the workshop. The *fair* ratings were justified with the provision of high-level information for architecture, but not enough detail for implementation. Similarly, the *poor* ratings were justified by the video providing a global understanding of what is needed, but lacking crisp information of the scope.

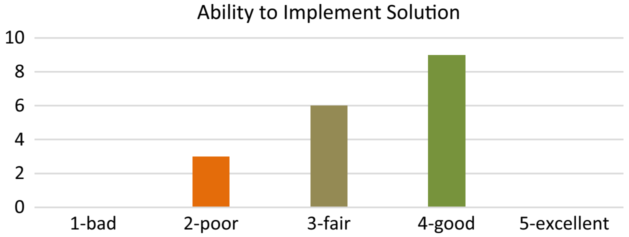

Fig. 5 Perceived ability of observers to implement the software that was discussed in the video

4.2.3 *Positive and negative aspects of the workshop video*

We asked observers to judge what was positive and what was negative with the workshop video when used for requirements communication. The positive judgments made by observers indicate strengths that should be retained. The negative judgments indicate needs for improvement.

4.2.3.1 Workshop video format Observers appreciated the presence of the real stakeholders in the workshop and the shared understanding achieved with the role-play and the stakeholder discussions about the role-play experience. The presence of real stakeholders generated trust and led observers to establish a relationship with them. However, the role-play approach to exploring requirements was also criticized. It led to scattering of information across the workshop, to lack of clarity whether requirements that were stated late would override requirements stated earlier, and to insufficient detail for some requirements. Also, apparent "illegitimacy" (observer's term) of a stakeholder led to critique. Finally, the absence of observers from the workshop was criticized because some of their questions were not answered. They obviously could not interact with the stakeholders. Table 1 gives an overview of the observer comments.

4.2.3.2 Workshop video contents When reflecting on requirements communication, observers commented the usefulness of the workshop video as an input to development. They acknowledged that a video had the specific advantage of capturing a requirements workshop as a primary source. It avoids the potential bias that the authoring of a requirements document would introduce. Also, the video captured role-play of prototype use and thereby captured rich information in an understandable manner. These advantages came with trade-offs, however. Irrelevant information was documented, some terminology used

Table 1 Format-related themes raised by observers

Theme	Positive	Negative
Stakeholders	*The parts when stakeholder R is describing how […] and the parts where stakeholders R and A discussed […]*	*Stakeholder P not legitimate*
Trust	*Stakeholders state their needs and expectations. Facial expressions and body language tell the real thoughts*	–
Relationship	*Developers see stakeholders and learn how to deal with them*	–
Understandability	*Play of scenarios of how to use the system to carry out processes, the explanations, and the real-world examples easy to understand*	*Information missed or misunderstood because scattered. People speak at the same time and about irrelevant things*
Clarification	*Requirements are introduced, questions answered, and decisions confirmed*	*Developers not part in discussions, thus many questions not clarified*
Interpretation	*The interaction between the users captures the "why" of the requirements very effectively. The stakeholders are actively discussing why certain requirements are important to them*	*[The workshop] can be interpreted differently by different people. Dialogue not enough to point out all requirements in detail*
Agreement	*Consensus between stakeholders*	*Some statements are ideas. Authentic, concrete, and agreed requirements missing. Some stakeholders do not contribute*
Configuration Management	–	*Hard to keep track of requirements as sessions may modify them*

by the stakeholders was not understood, and a structured overview of "what to do" was missing. Table 2 gives an overview of the comments.

4.2.3.3 Video consumption Observers complained that video watching was time-consuming, and the right information could not be found easily. At the same time, however, they acknowledged the value of replay that allows them to watch a video again for achieving requirements understanding and discovering information they had previously overlooked. Further, storage space and bandwidth were critical for a good video-watching experience. Table 3 gives an overview.

Comparable with the comments about the workshop, observers expected the video to be recorded, edited, and provided with high quality. The recording of the presented video had quality problems, however, which were easily discovered and criticized. A second reason for critique was the effort needed to browse and search inside a video. Support was desired by observers to improve the accessibility to the video contents.

4.2.3.4 Sufficiency for requirements communication Observers disagreed about the sufficiency of workshop videos for requirements communication. Some focused on the start of the development work and judged the video to be a good source that provides enough information for an architect to start working. Others felt bored or were concerned about the need for clarification during development. They judged the video to be insufficient and suggested meetings and support by domain experts to resolve open questions. Table 4 gives an overview.

4.2.3.5 Side effects Observers also saw benefits in the workshop videos beyond their originally intended use for requirements communication. The video eased the learning of a new domain and introduced the observer to practical requirements engineering. These benefits were not achieved fully by the specific workshop video that was studied, however. For example, the introduction to the domain should have been complemented with a description of previously existing solutions. Table 5 gives an overview.

Table 2 Themes related to requirements documentation

Theme	Positive	Negative
Primary source	*No bias from retelling of goals and requirements*	*The role games include irrelevant information for an information system (chattering)*
Depth	*All details recorded, thus omissions avoided. Captures the "why" of requirements better than other formats*	*Not concrete: needs more precise "what to do"*
Understandability	*Understood most because of discussion summaries*	*Domain-specific terms with many possible meanings used. Jargons used should have been explained. What is N3?*

Table 3 Themes related to video consumption

Theme	Positive	Negative
Watching	*Watch again*	*Time-consuming to familiarize, watch, and analyze important parts*
Searching	*Replay to get missed important information*	*Search requires watching whole sequence*
Understanding	–	*Video to be watched back and forth to catch everything and solve use cases puzzle*
Storage	–	*Space-consuming*
Bandwidth	–	*100 kB/s too little*

Table 4 Themes related to development input

Theme	Positive	Negative
Usefulness	*Good source. Overall good and helpful for developer. Mostly interesting observer and video*	*Bored watching the video*
Sufficiency	*Enough requirements and boundaries suitable for architect to start*	*Other meetings and support by domain experts needed to resolve open questions*
Complements	*Good complement for a vision statement*	*Vision and video insufficient*

Overall, the judgment of the video as a means for requirements communication confirmed the view that workshop videos are interesting to use in situations where previously only requirements specifications were used. In addition to achieving a requirements understanding that is good enough to start development work, they bring efficiency, introduce people to a new domain, and support requirements engineering training. To be effective, however, workshop videos need to be created and edited to contain only relevant and at the same time all necessary information. Also, they need to be enhanced with complements that makes the video contents understood and with support to clarify questions. One of the important aspects to address is how a correct understanding of specialized terms can be fostered in a situation where stakeholders and developers cannot ask questions to test such understanding.

These generic findings on video handling confirm findings of others, e.g., Heath et al. [10]. Future research should explore the delicate balance between naïve and simple recording versus slightly more professional handling of videos at a slightly higher cost and effort. As Fig. 3 shows, developers are able to identify requirements and could be involved as a crowd that helps interpret and index a video.

4.3 Workshop video quality

4.3.1 Noteworthy events observed in the workshop video

We wanted to know what the observed events were that affected the perceived usefulness of the video. According to the markers, the study participants complained about problems in the workshop and gave positive feedback about successful moderation by the requirements engineer. Figure 6 shows the aggregated feedback about these factors.

Remarkable peaks are labeled with *P* if they denote problems in the workshop. Six such problems were identified.

P1: Many complaints about noise, difficulty to understand due to parallel conversations and a person entering the room.

P2: Confusion due to parallel conversations.

Table 5 Themes related to side effects

Theme	Positive	Negative
Production	*Saves money and energy, thus uses resources efficiently*	*Requires extra skills and resources for filming and video editing*
Introduction to domain	*Video eases the learning. I have no experience in the development of such systems. The video was valuable to gain knowledge of it*	*I am missing a description on previously existing solutions (if any)*
RE training	*Good to see practical requirements engineering*	–

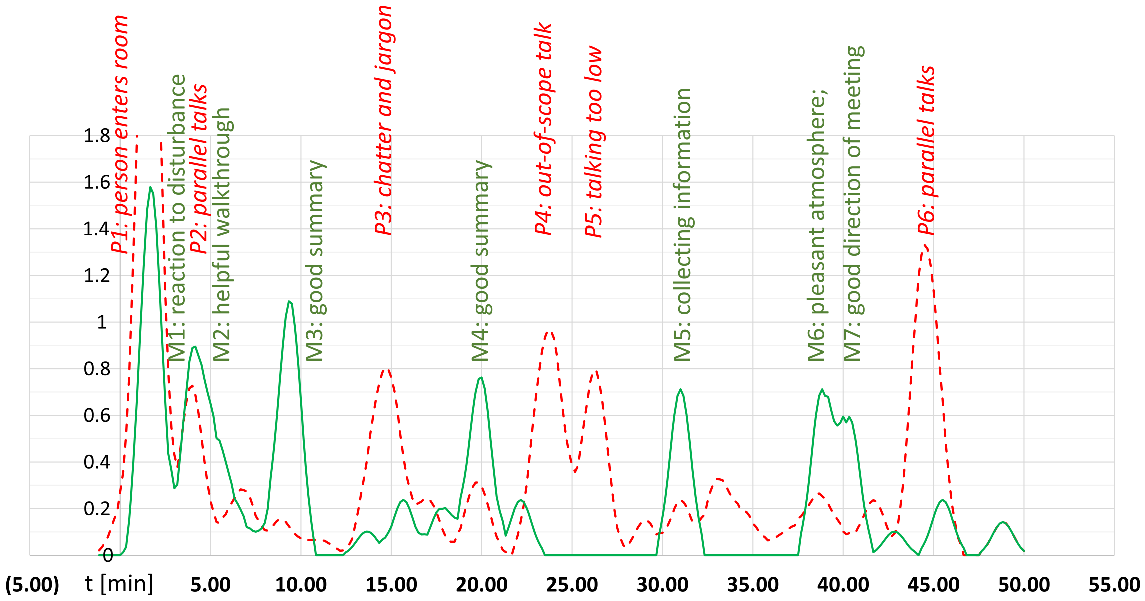

Fig. 6 Workshop-related factors that influenced video usefulness: interventions of the requirements engineer or film crew (*solid green*) in relation to workshop problems (*dashed red*)

P3: Some “chatter” and “irrelevant conversation” is mentioned in the feedback. Jargon was used and not explained. At this point, very few requirements are communicated, which triggered several complaints.

P4: Confusion due to parallel conversations that were out of scope (irrelevant) for the project.

P5: A workshop participant talks about how pharmacists interact with the customer. Several observers complained about him speaking too low. They did not understand what was in scope and what not.

P6: Observers find it difficult to concentrate on so many things at the same time. Parallel discussions among stakeholders; the camera does not always focus on the relevant discussion.

After the initial peak, complaints dropped drastically to less than half of the initial rate. Also, there was an extended period covering M5–M7 that was without peaks of problems beyond the upper quartile level.

Problems were avoided or mitigated with moderation by the requirements engineer. Remarkable peaks are labeled with *M* if they refer to such moderation. Seven moderation-related peaks were visible.

- Observers mostly referred to the requirements engineer summarizing use cases or complex interactions (*M3*, *M4*, and *M5*) and the fact that walk-throughs helped them to understand use cases and processes better (*M2* and *M7*).
- Other remarkable triggers for positive feedback were the reaction of the requirements engineer to the person entering the room late (*M1*), which had caused confusion and complaints (*P1*). The requirements engineer restored a working atmosphere and repeated what had been said before.
- Just before this intervention of the requirements engineer at *M1*, the video recording had been paused by the film crew for a short time—the only cut in the entire video. Observers had noticed this and mentioned it in positive feedback.
- There was also a comment about the “pleasant atmosphere” of the workshop discussion (*M6*). Thus, positive feedback referred to the video and to the moderation of the workshop.

4.3.2 Perceived satisfaction with the workshop video

Altogether, observers stated moderate satisfaction with the workshop video. Figure 7 shows the distribution of the opinion scores that were obtained from asking observers “how satisfied are you with the video?” The mode and median satisfaction scores of the answers were *good*. Two observers perceived the video to be *excellent*. Although no participant judged the video to be *bad* or *poor*, a large

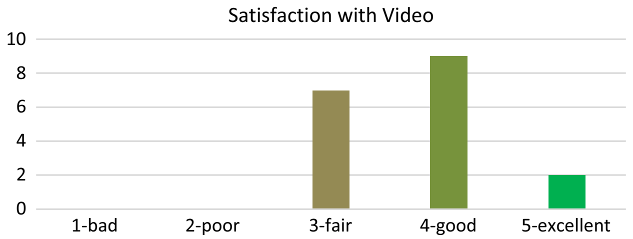

Fig. 7 Counts of answers related to observers’ satisfaction with the workshop video

Table 6 Domain-related themes

Theme	Positive	Negative
Context	*[…] idea of where the system is supposed to operate in and who are the users […]*	–
Process	*The video shows the process […] that should be carried out*	*How is [process step x] going to function?*
Devices	–	*A summary of each device telling where it is to be used, who will use it, and a short description of its interfaces would help*

Table 7 Stakeholder-related themes

Theme	Positive	Negative
Goals	*Stakeholder R explaining what he wants to accomplish […] and to know exactly the user needs*	–
Expectations	*Wants expressed and explained*	*Some wants unclear. Expectations imprecise*
Rationales	–	*Why should the functionality x be taken out [and be replaced by] very repetitive and error-prone task for the user P?*

minority considered it *fair* and thus implied that there was considerable room for improvement.

The *excellent* ratings were justified with seeing the real stakeholders discuss the requirements, which was considered interesting. The *good* ratings were justified by the good quality and interestingness of the video and by the requirements understanding it generated. Concerns were raised about the relevance or quality of selected parts of the video, the omission of some requirement types, and the difficulty of navigating within the video. The *fair* ratings were justified by similar positive arguments. However, many concerns were raised about the lack of structured moderation, the lengthiness and quality problems of the video, and jargons used by the stakeholders in the workshop.

4.3.3 Factors affecting perceived usefulness

We asked observers to state the factors that affected their perception of video usefulness. The question was intended to collect feedback that is useful for guiding the production of future workshop videos. Positive judgments made by observers indicate strengths that should be retained. Negative judgments indicate areas that need improvement.

Obviously, there are several levels that have an impact on the evaluation. According to the collected feedback, the levels reach from workshop content over workshop moderation to specific aspects of using video for documentation. Each level builds on and is constrained by all lower levels. A poorly prepared workshop cannot convey requirements effectively—no matter whether it is video-recorded or not. A workshop video can only unfold its fullest potential if these factors are controlled.

4.3.3.1 Requirements discussed in the workshop When reflecting on the requirements discussed in the workshop, observers commented on the system, its domain, stakeholders, and the project in which the system will be built.

Observers appreciated that the video introduced them to the context and processes that the system will be used in. However, they would have preferred a systematic and complete introduction to the process and the devices the system interacts with. Table 6 gives an overview of the observer comments.

Observers appreciated that the video described stakeholder goals and expectations. However, they would have wished more clarity and precision in how stakeholders expressed their expectations. In particular, rationales should be stated when expectations are not intuitive for the recipient of the video. Table 7 gives an overview of the comments.

Most of the themes commented by observers concerned the system. They appreciated the overview of the system in terms of scope, scenarios, features, functions, quality, and interfaces to be supported and criticized omissions or imprecisions here. Some observers would have wished more detail, while others expected the details to be elaborated at a later stage in the development process. Also, data formats, constraints such as for the memory, and target platforms were expected to be stated. Table 8 gives an overview of the comments.

Observers did not like that the video omitted project-related themes. They missed information about why the system is needed, constraints for planning and budgeting the project, and inputs for risk management. Table 9 shows the comments.

Overall, observers expected all aspects of requirements to be discussed that otherwise would be specified in a

Table 8 System-related themes

Theme	Positive	Negative
Scope	*[I liked] the defining moments for deciding the scope of the system [and knowing the] use cases that are out of scope*	*[The] scope of system, legacy, features, and "what to do" [were] unclear. I do not know [what parts] should be made or not, or whether they should be made flexible in order to adapt to the process*
Scenarios	*The most valuable parts were where stakeholders R and A were explaining the user stories. The scenarios played out gave a lot of insight*	–
Features		*I have not gotten any details of the features [x, y, z]*
Functions	*Because the overall functionality is described […], lower-level details can be built upon it*	*[The] requirements [were] imprecise. Some functions were not discussed or ambiguous*
Quality	–	*[The] quality [is] unclear. Quality requirements were not discussed*
Interfaces	*[The description of] the different interfaces required for [function T, RI] by the users [D and P] was good*	*The system will depend so much on the device BR, [but the] interfaces for integration of the BR device were unclear*
Data and I/O	*I was interested in the information, input, and output I have to provide in the system*	*I am missing […] the format at the structure of the data*
Constraints	*Needs and discussion of assumptions and constraints reasonable*	*It would have been good to know […] the memory constraints and any other constraints*
Platform	*Hardware with enough detail. System feasible with current technology*	*Operating systems unclear. It would have been good to know what platform the product should be implemented on […]*

Table 9 Project-related themes

Theme	Positive	Negative
Rationales	–	*Background story of the system and of the customer needs missing*
Planning	–	*The project was not discussed. The main parts missing are […] the deadlines*
Budget	–	*The accurate budget to make sure whether the customer can accept [the project]*
Risk	–	*There is no description of project risk […for] the software developers and users of the product*

requirements specification. This includes the system context, stakeholders, and requirements for the system. Our results show that this was not enough, however. We were surprised by the importance of project-related themes that extend the information that would be stated in a requirements specification. Another important result relates to system design. Although the discussed system exhibited important user interfaces, no one complained that the details about user interaction design and the structure and appearance of the user interface appeared were not addressed in the video.

4.3.3.2 Workshop moderation Noise, distraction, or inappropriate behavior of participants can diminish the effectiveness of the workshop. In that case, a video can only convey those requirements that surfaced in the workshop.

Observers did not like that the workshop was interrupted by disturbances. They would have wished the stakeholders to be present on time, well prepared, and more disciplined. Also, the discussions should have been better focused on the essence of the relevant parts. Table 10 shows the comments.

Observers judged that the workshop was good enough. Positive remarks were given for letting stakeholders experience the product and for summaries that were made by the requirements engineer to check for a shared understanding of requirements. However, a lot of negative feedback was given about the moderation. The moderator appeared to lack confidence. When watching the video, the meeting appeared to be started without agenda and without introduction of participants, the workshop to be without structure, and the workshop participants not managed. Observers also asked for better-structured explanations, more summaries, and a white board to show the shared understanding developed during the workshop. Table 11 gives an overview of the observer comments.

4.3.3.3 Video production When reflecting on the workshop video, observers commented important lifecycle stages of the video, including recording, production, and consumption.

Table 10 Disturbance-related themes

Theme	Positive	Negative
Latecomers	–	*Some parts of the video are distracting, for example when there is one person who was late and disturbing the discussion*
Difficulties in Prototype use	–	*Sometimes, scenarios were interrupted for someone pressing the wrong button on the device BR, which did not really matter*
Simultaneous talking	–	*[Not useful were] the parts where all the people were speaking*
Irrelevant parts	–	*The part where they were discussing if x should be delivered in a box or a bag is not really relevant*
Lengthy parts	–	*Some of the discussion could get a bit lengthy, especially when there was a discussion on whether a certain requirement was in scope or not*

Table 11 Moderation-related themes

Theme	Positive	Negative
Moderator	–	*Moderator lacked confidence*
Start	–	*Agenda not discussed and participants not introduced. There may be a need for a short introduction about the number of team members before the meeting*
Scene	*Letting customers experience the product*	*No whiteboard to show and break understanding down*
Structure	*Process became clear and inconsistencies [in the process] detected*	*The facilitator should take the workshop in technical manner: explain the […] system and stakeholders*
Participation	*Good-enough workshop where all had a role and got involved. All stakeholders involved. The architect was attentive*	*Some were very much involved and dominated the discussion. Some were partially involved, inactive, or not attentive. Development lead [= architect] should be more active*
Explanations	–	*A more detailed and structured explanation of the system itself […] and what they want to happen between A and B*
Summaries	*The most valuable parts were when features were summarized and clarified to check for a shared understanding with everyone in the room […]. Knowing this much is sufficient for a developer*	*At some point may be a recapitulation of all the requirements: it will be easier than watch the video again*

Observers disagreed with each other when judging the quality of the video recording. Some stated the video was well recorded and with not too much noise, while others said it had technical problems. They included poor camera movement that hindered capture of all facial expressions, as well as too low and too fast speaking. Table 12 gives an overview of the comments.

Observers reported that the video covered the important discussions well, while at the same time stating that it was too long. It contained parts that were ambiguous, awkward, irrelevant, and out of scope. These parts should be shortened or removed during editing of the video. Table 13 gives an overview of the comments.

Overall, observers confirmed that the workshop video was an adequate format for communicating requirements. At the same time, the critique was expressed about the fact that participants were learning and that the requirements engineer experienced surprises that had to be moderated. Although the observer expectations were rather high, many problems could have been avoided at acceptable cost and effort. Selection of only trustworthy participants, adequate preparation of the workshop, strict structuring of the sessions and of the discussions, and repeated summarization and visualization of agreements would have enhanced the experience of those who watched the workshop video.

It should be noted that our lightweight approach is meant to be applied by regular projects with a moderate ability and limited resources. Therefore, deficits may occur at all levels. In our study, we wanted to conduct a reasonable requirements workshop and see how video-recorded would be evaluated at all levels. Therefore, we selected a workshop video that was not perfect, but represented practice in a realistic industrial setting. The imperfections of the workshop video allowed us to evaluate all intertwined levels in this exploratory case study.

4.4 Recommendations for requirements communication with workshop videos

4.4.1 Intention to use

The majority of observers would use workshop videos in their own practice. Figure 8 shows the distribution of answers obtained from asking observers “would you use a

Table 12 Themes related to video recording

Theme	Positive	Negative
Quality	*Well-recorded, good-quality video*	*Low quality with recording problems*
Camera	–	*Just one camera with poor movement. Some facial expressions not captured*
Voice	*Not too much noise*	*Some spoke too low and too fast*

Table 13 Themes related to video editing

Theme	Positive	Negative
Coverage	*Good coverage of important discussions*	*Too long with ambiguous details and lost and awkward parts*
Irrelevancies	–	*Irrelevant details. The parts not speaking about the system operation are not really useful*
Redundancies	–	*The parts where the discussion continued on into features that had already been clarified and declared out of scope felt unnecessary*

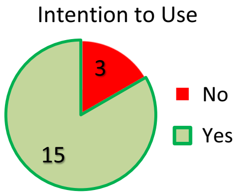

Fig. 8 Counts of observers' intention to use workshop videos for requirements communication

video for requirements communication?" 83 % would do so, while 17 % would not.

The *positive* voices saw great potential in the workshop videos. According to them, such videos capture detailed requirements, save time and money for requirements engineering, and can be used to reach many recipients. The videos would be used to obtain information, to clarify and resolve ambiguity, and to brainstorm and take decisions about the product to be developed. The videos should contain role-play by real clients, be short, and support extraction of useful information. The videos would be used in combination with written requirements and follow-up meetings with stakeholders. However, developers cannot provide input or feedback when watching the video. It was proposed to have some developers participate in the workshop in order to overcome this limitation.

The *negative* voices criticized the workshop video contents as being sketchy and incomplete, thus insufficient for development. Workshop videos were judged to be immature, to depend heavily on the requirements engineer, and to be hard to use for extracting valid requirements information. Workshop videos should be used to document a workshop, but not to replace a final requirements document or a personal meeting.

Observers' intention to use a workshop video for requirements communication matched their perceived usefulness of the workshop video, but not the satisfaction with the video. Those who would use a video judged the usefulness of the video higher than the observers who would not use it. The positive judgments also matched high perceived ability to implement the solution. Thus, increasing the usefulness of workshop videos for requirements communication is likely to increase the intention to use them of that purpose.

In contrast, there was no difference in the judgment of the video quality between those who would use a workshop video for requirements communication and those who would not. Based on the presented results, we conclude that the form of the video and how it was presented were good enough. Improvement of the video might have increased satisfaction with the video, but not the intention to use the video.

4.4.2 Recommendations for workshop video use

When asked how they would use workshop videos, observers suggested two different types of workshop video purposes and proposed five uses for them.

Workshop videos may capture two different activities in a project: requirements engineering and design. This study explored a *requirements* workshop. Observers expected such requirements workshop videos to provide global information about the software product, the project goals, and the stakeholder expectations. Alternatively, *design* workshops may be recorded. These videos should have a technical focus and capture how to achieve the project goals.

Use for team member introduction: Observers would use workshop videos to introduce the team to the project, as a source for requirements elicitation, as a support for inquiry, and as a reference. To get introduced to the system and the domain, a team member would watch a workshop video once. It was judged to be good for the team to see the whole video to understand the workflows and develop a feel for how users will use the system.

Table 14 Workshop preparation

Aspect	Recommendation
Knowledge	*Know the problem domain and workflows to be supported by the software and obtain a paper story pre-workshop*
Genuine stakeholders	*Bring a genuine may be client of the application [to the workshop]*

Table 15 Moderation of the workshop

Aspect	Recommendation
Introduction	*Start by introducing workshop, participants, and roles*
Discussion management	*Allocate and limit time for everyone to speak, motivate and fully involve all participants, and prevent someone from dominating with unnecessary out-of-scope talks*
Discussion support	*Use a blackboard to sketch the workflow, use cases, ideas, requirements, and comments and build a shared dictionary for domain-specific terms*
Closure	*End the discussion by summarizing the scenarios and system features*

Use for requirements understanding: Workshop videos would be watched to understand stakeholder expectations, requirements, and what is going on in the required features. In particular, they should convey how the system should behave in specific situations and the functionality needed to implement the features. Workshop videos might need to be watched repeatedly for this purpose because requirements may not be obvious initially.

Use as a support for requirements inquiry: Workshop videos would be watched as a support for requirements inquiry. The video should be used as a basis to ask well-informed questions to domain experts later on.

Use as a reference: Workshop videos, finally, would be used as a reference for remembering discussions and agreements when needed. Thus, the video is considered part of documentation. The use of workshop videos would avoid contacting stakeholders for clarification unnecessarily when questions emerge during development. In particular, they would be used to reflect on how to implement the solution, to search for things that were missed or uncertain in the requirements document, and for reviewing the implementation from the perspective of the stakeholders.

Use for training: The videos show real-life workshops with requirements engineers in action. Study and discussion of the video are considered useful for education of requirements engineers.

Overall, the breadth of contents and uses proposed by observers suggests that workshop videos have a wider potential scope of application than we initially foresaw. Our results indicate that they are useful to capture situations where project members and stakeholders take decisions that are disseminated to people who implement the decisions, but could not participate in the decision making. Recipients would use the videos for self-education and for reducing their dependency on stakeholders.

4.4.3 Recommendations for workshop video production

4.4.3.1 Recommendations for requirements engineer When asked for recommendations on how to improve the requirements workshop, observers recommended that the requirements engineer improves the preparation, moderation, and requirements engineering for the workshop.

Observers felt that the workshop preparation should be improved. To appear more confident and to better structure the workshop, the moderating requirements engineer would have to know the product and domain. To improve the credibility of the workshop, only genuine stakeholders should participate. Table 14 gives an overview of the quotes.

Observers felt that the moderation of the workshop should be improved. The workshop should be framed by an introduction and a closure session. To enable the developer benefit from them, these sessions should appear as scenes in the workshop video. During the workshop, the discussions should be better managed, and the results captured visually. Table 15 shows the quotes.

Observers felt that requirements engineering practices should be improved. A business case should be developed during the workshop that balances business and customer needs with the project constraints. Scope should be sharpened, statements disambiguated, and decisions critically reviewed to ensure their validity. Table 16 gives an overview.

Overall, many recommendations were implemented already, but not used systematically. Also, the workshop was started and closed as observers suggested, but these parts had not been recorded. To address these weaknesses, the requirements engineer should follow a strict protocol that helps implement the recommendations. To win the support of the workshop participants, the protocol should

Table 16 Requirements engineering during workshop

Aspect	Recommendation
Business case	*Combine business and customer needs and balance them with capacity and time by prioritizing requirements*
System scoping	*Remove conflicts and agree on scope with all stakeholders*
Scenario scoping	*State start, play out the use case, end with a brief, and clear summary that details the steps and parameters*
Clarification	*Ask questions when something is blurry and make things clear by repeating what stakeholders said*
On-the-fly validation	*Summarize requirements repeatedly, identify and show differences between real-world practices and stakeholder-stated workflows, obtain feedback on the understanding, and track requirements changes*

Table 17 Video equipment

Aspect	Recommendation
Sound	*Allow the participants to be heard clearly by giving each a good microphone, by doing sound check, and by reducing background noise*
Video	*Capture multiple perspectives; stabilize the image; avoid dizziness due to camera moves; use multiple good cameras that are fixed or fix a camera on a tripod with wheels*

be agreed with them. The study results show that a complete recording of a disciplined workshop would improve the experience of the developer who is watching the workshop video. In particular, the introduction and closure sessions and the continuous checking of shared understanding are important for the developer to understand the objectives and the scope of the system.

4.4.3.2 Recommendations for video crew When asked for recommendations about how to improve workshop videos, observers recommended that the video crew uses good equipment in sufficient number, record overview and detail without disturbances, and edits the video to add meta-information and remove problems.

Observers suggested the use of high-quality sound and video recording devices. These devices should be used at important locations to ensure crisp capture of voice and professional pictures. As an alternative to the use of multiple devices, TV studio equipment should be used such as a camera tripod with wheels. Table 17 shows the quotes.

Observers recommended more systematic capture of the workshop scene, of the discussions that are ongoing, and of the artifacts that used by the stakeholders during the discussions. The camera moves should provide the person watching the video with an overview of what is happening and with sufficient detail to understand what is said and to know what is manipulated. Also, low-quality voice and disturbing background should be avoided. It is remarkable that the audio problems were reported much more frequently than visual problems. Table 18 gives an overview.

Observers suggested edits and complements to the video that improve the quality of the video and the accessibility of its contents for the person who is using the video. Distractions and noise should be removed, and a good flow of the scenes created by cutting the video appropriately. Subtitles and meta-information should be added to understand the themes of the scenes. A transcript should be created, and an index provided to ease search and navigation. Table 19 gives an overview.

Overall, the recommendations can increase the quality of workshop videos. However, many recommendations are in conflict with our original intention of the workshop videos. Readily available equipment such as amateur cameras should be used for recording a workshop and requirements for video editing equipment and effort be minimized. This contradiction shows the trade-off when the workshop video technique is used in practice. In some situations, practicability is critical. In other circumstances, quality is more important. For example, the use of the workshop video for dissemination and publicity purposes in addition to requirements communication may call for high-end equipment and professional recording and editing.

4.4.3.3 Recommendations for hand-off to developers When asked for recommendations about how to improve the hand-off of the workshop video to developers, observers recommended providing documentation of the workshop and of the requirements in addition to the video and to follow-up the hand-off.

Observers recommended documenting the workshop with a report about information about the workshop participants and with a summary of the workshop results. In addition, the devices and artifacts used during the workshop should be made available. A glossary should be used to clarify terminology. Table 20 gives an overview of the recommendations.

Observers saw workshop videos as a complement to written specifications. They recommended

Table 18 Video recording

Aspect	Recommendation
Overview	*Start video by rotating through the room while the requirements engineer introduces all stakeholders*
Discussion	*Focus on the talking stakeholder, including lips and hands, capture all stakeholders in across-the-table discussions, and capture all face reactions*
Interaction	*Capture details by zooming artifacts and user interfaces used to demonstrate system functionality. Hold to focus*
Audibility	*Avoid unclear requirements by recording the voice of the speaker loud enough. Avoid irrelevant competing discussions, laughter, and noises when people speak*
Background	*Avoid movement of people without any meaning, also outside the room, that divert attention*

Table 19 Video editing

Aspect	Recommendation
Distractions	*Remove distracting parts. Cut scenes where everybody is speaking at the same time or that are irrelevant. [However,] avoid [a cut] when a person is not finished [talking]*
Noise	*Do voice processing for reducing noise, avoid differing sound levels, and remove noisy parts in the video*
Subtitles	*Subtitles of the conversation would clarify*
Meta-information	*Add labels or markers in the time line to show important parts in the video. Add subtitles, perhaps on a black screen, to indicate features*
Transcript	*A transcript would make it easier to know which part of the video to skip to if it needs to be watched again*
Index	*Allows to go back and clarify doubts*

Table 20 Workshop documentation

Aspect	Recommendation
Report	*Should contain the minutes of the meeting or notes taken during or after the workshop that summarizes the discussions and the requirements that were accepted by all stakeholders. A report would allow developers to refer to the meeting and verify whether they have missed something in the video*
Contact list	*Should contain the contact details of each person. A contact list would allow asking for more details if there is something not understood in the video*
Summary of results	*Should contain the requirements that were accepted by all stakeholders and a clarification of all expected features. These results may be represented with a mind map, use case or class diagrams, or use case or scenario descriptions. A summary would clarify all expected features, boost developer confidence, and help remember all features that were agreed upon*
Devices	*The devices used for prototyping should be made available together with their complete documentation. Availability of the devices would give developers an idea of the device appearance*
Artifacts	*The artifacts that were discussed during the workshop, such as documents and forms, should be made available. They would help developers understand the discussion more clearly*
Glossary	*Should contain a clear explanation of terminology and complex words that might have multiple meanings. A glossary would help developers discovering wrong assumptions and understanding*

- using a vision statement or document to set the objectives for the project,
- using descriptions of existing solutions, assumptions, and constraints to guide development, and
- using conceptual models, a detailed SRS, a supplementary requirements specification, and mock-ups to provide specification details.

Table 21 gives an overview of the detailed statements.

Observers recommended follow-up of the workshop video hand-off. Meetings between the developers and the requirements engineer should be held to provide clarification to questions. Complementing documentation should be provided. Workshops should be planned to allow developers asking questions to stakeholders and handshaking implementation proposals. Table 22 gives an overview.

Overall, the recommendations describe how workshop videos should be embedded in the development process. Workshop videos should be used as a means to educate and inform developers. They should not substitute notes for reporting about the workshop or a requirements catalogue for managing and tracking development. Correct use considers the specific strengths and weaknesses of the technique. The example of the workshop video analyzed in this study, the strengths and weaknesses identified by the

Table 21 Requirements documentation

Aspect	Recommendation
Vision statement	*Contained the problem statement and the idea and scope of the solution. The vision statement helped to understand the video much better than just watching the video*
Vision document	*Should define the system actors and workflows. A vision document would make it easier to understand the entire system during development*
Existing solutions	*Examples of existing solutions similar to the discussed system. They would allow seeing how competitors have solved similar problems and help starting solution development*
Assumptions and constraints	*Assumptions about users, runtime environment, and development work and project constraints about time, funding, personnel, equipment, and geographic location should be stated. This information would allow understanding the factors that influenced the requirements and help the stakeholders define reasonable needs*
Conceptual models	*Should contain UML diagrams such as use case and activity diagrams, use cases with possible user interactions. Such requirements that document all wanted details of the proposed system would improve the confidence of the developer.*
Detailed SRS	*Should be a conventional requirements document with text, diagram, sketches, and drawings to describe requirements in more technical detail. The SRS should make clear what the final requirements are, capture all specification detail in a searchable format, and make it easy to track requirements. An SRS should be written by the requirements engineer during or after the workshop to make sure that all requirements are captured in detail*
Supplementary requirements	*Should contain quality requirements and data formats, structures, and examples*
Mock-ups	*Should specify the layout of the user interfaces. User interface screens should help when starting solution development*

Table 22 Follow-up

Aspect	Recommendation
Clarification	*Hold a requirements meeting with team members to allow developers understand customer requirements*
Other documents	*[Other inputs for development, such as] hardware manuals and [deliverables from development such as] architecture*
Workshops	*Meetings would be used to get rich exchange with stakeholders where developers ask questions, suggest and sketch implementation, get additional information, and develop a shared vision*

observers, and the recommendations to improve the technique reported in this paper provide the insights necessary to make workshop videos effective.

5 Validity

The first author of this paper was requirements engineering moderator in the recorded workshop and conducted the class in which the video was evaluated. The key threat to validity of the presented study is thus respondent bias of the students who participated in the study. The responses risked to be friendly. We had introduced measures to prevent respondent bias and have compared the results with a replication of the study inside the concerned software project. The replication was performed with the project's head designer who entered the project when the requirements engineer left the project. The co-authors of the study were independent researchers who ensured the absence of researcher bias through triangulation and evaluated the obtained results for the presence of respondent bias. This section describes in detail the validating replication, the actions used to limit threats to validity, and the examination of bias in the presented results.

5.1 In-project validation with head designer

We have validated the obtained results by applying the presented study protocol with the head designer of the application that was discussed in the workshop video. The validation helped us to evaluate external validity: Whether the results obtained with experienced software engineering student observers can be transferred to industrial practice.

Both the head designer and observers had software engineering education. Yet, the profile of the head designer differed in two aspects. In contrast to the student observers, he had experience in developing the kind system that was specified. The specific context for which the system was

Table 23 Judgments of the head designer

Indicator	Judgment
Utility for requirements communication	2—good enough
Ability to implement solution	3—fair
Satisfaction with workshop video	4—good
Intention to use	Yes

developed and one of the central features were new to him, however. Also, in contrast to the other observers, he had a genuine interest in understanding the requirements because he was expected to deliver the system.

The feedback of the head designer was consistent with the presented results from students. His ratings, summarized in Table 23, corresponded to the median ratings shown in Figs. 5, 6, 7, and 8. In his overall judgment of the workshop video, he stated two of the four uses that were proposed by the student observers already. He considered the workshop video to be a substitute for initial requirements that give an overview of the solution. Also, he would use the video to clarify the requirements that can be identified with the role-play.

Also, the head designer's evaluation of the strengths and weaknesses of the workshop video was consistent with the student observers' evaluation. He perceived the workshop format advantageous because the recording of the stakeholders gave insights into the sources of the requirements and the body language and discussions pointed to uncertainties. The summaries given by the requirements engineer clarified key requirements and anything that was not correct. The video allowed to return to the scenes as many times as necessary to get a better understanding of the requirements and business rules. Finally, the video documented the requirements more accurately than a workshop summary would have done.

Also, his critiques were consistent with the critiques given by the student observers. He criticized that the format of the recorded workshop made requirements understanding difficult because it did not separate the discussions of functionality, legacy, and proposed solution. Some parts of the system discussed in the workshop could have been clarified better. Also, he complained about workshop disturbances. Some people in the room did not contribute and were a distraction. For example, they were discussing over each other or within groups who spoke among themselves. Laughs, banter, and mishaps distracted and may have flagged up important issues. Finally, he would have appreciated seeing the artifacts to be used or automated by the software to be part of the workshop documentation.

Also, congruent with the student observer feedback, the head designer had an ambivalent impression of the requirements discussed in the workshop. He saw the advantages and disadvantages shown in Table 24.

The head designer identified two issues that were not discovered by the other observers. Opposed to the student observers, he understood that N3 was a third party who needed to be integrated. He signaled the lack of access to that third-party to be an issue that needs resolution. Also, he mistrusted cuts in the video. Important information may have been skipped because it seemed irrelevant during editing.

Some critiques from the student observers were not confirmed by the head designer. He did not report that the moderator would lack confidence. Also, he did not state video recording quality was low or that the video would have been difficult to consume.

The overall accord of the head designer's judgment with the presented study results indicates that observers' lack of experience with the solution and the lack of genuine interest in the requirements did not substantially affect the judgment of the workshop video and its use for requirements communication. Still, some differences remained that can be traced to his work experience and his connection with the software project. For the features that he had experience with, he expected detailed information about functions, quality, interfaces, and data. For the innovative feature where he lacked the experience, he appreciated the level of detail produced with the recorded role-play. Also, he accepted the challenges of real-world requirements engineering and saw less problems with workshop moderation and video quality than the student observers. Finally, his connection with the project helped to reduce understandability problems and disambiguate apparent jargon.

To use workshop videos for requirements communication, the head designer gave the following recommendations. He suggested that the video should inform the observer about the requirements and not illustrate the learning of the stakeholders. To achieve this aim, he stated that the video should make clear how the as-is and the to-be processes are demarcated and what the dependencies of the new system with the legacy processes and system are. Also, the video should separate requirements topics by first focusing on the role-play of functional requirements and then on the discussion of non-functional requirements.

He expected that the workshop video would be complemented with the following documentation. An as-is analysis should specify the existing processes, including workflows and source documents that would be used or automated by the system. A vision document should

Table 24 Judgment of requirements by head designer

Theme	Positive	Negative: missing or unclear
Process	–	*System administration*
Goals	–	*Comparison of the as-is situation with the to-be situation*
Rationales	–	*Rationales for certain actions*
Scope	–	*Boundaries of the solution*
Functions	*When and how the innovative feature is used*	*Business rules*
Quality	*Number of items to be handled by the system, login*	*Frequency of system use, data retention length, user control, access levels, login methods*
Interfaces	–	*Interfaces to existing systems*
Data and I/O	*Data-related requirements*	*Type of data to be categorized and the channels to input*
Platform	*The system being used*	*The system being used*

describe the vision, stakeholders, purpose, and features of the new system.

5.2 Analysis of threats to validity

The presented study contributed with an evaluation of how a requirements workshop video is perceived and would be used for requirements communication from the perspective of developers. The study analyzed rich feedback about the video that was collected from a substantial number of people. The mostly qualitative nature of the multi-observer single-case research design shares many of the threats to the validity of empirical case study research. We therefore use the classification scheme proposed by to discuss the threats to validity [66, 67].

Construct validity reflects the extent to which the operational measures represent the concepts that have been investigated according to the research questions. The biggest risks of the presented research are as follows. Observers may have answered the questionnaire without actually having watched the video, the questions provided in the questionnaire may not have been understood, and the interpretations of researcher and subjects may have been inconsistent. As a result, the presented judgments and recommendations may be irrelevant for the purpose of the research.

To ensure that actual experience of watching the workshop video was reported, the study protocol required the observer to annotate the video. Each observer was incented to give broad coverage of the video with in-depth comments. To discover misunderstandings and diverging interpretations, each answer required by the questionnaire had to be complemented with a rationale. Again, observers were incented to provide in-depth rationales. The in situ video annotations gave insights into the thinking of observers while they watched the video. To avoid a double cognitive load of annotating and concurrent watching, the video could be stopped and easily spooled to the location of the annotation. The questionnaire contained questions that were only answerable when the video was watched. The a posteriori answering of the questionnaire gave the possibility to reflect on the experience and report a consolidated judgment.

Between six and 54, video annotations were made per observer. The first quartile was 15.5, and the median was 25. For one observer, we could not capture the annotations because of server problems. All rationales given for the annotations were understandable and plausible. We considered this extent of video annotation to be enough to trust that the video was watched. In addition, the annotation analyses shown in Figs. 2, 3, and 6 show that the feedback was rich enough to capture meaningful profiles of the video. Also, all questionnaires were completely answered, and the rationales given for the answers were understandable. Only in one case, we doubted the quality of some of the answers. Instead of judging the video, the respondent judged whether she would be able to contribute to a development team that would implement the software. Still, we judged that this view is valid for interpreting a workshop video from the perspective of a developer. The retaining of her answers contributed to a comprehensive picture of workshop video perception, which was the objective of the presented study.

The presented results were triangulated from 18 observers and the two data sources video annotations and questionnaire. Conflicting opinions discovered in the analysis, and synthesis of the collected data was made explicit by stating both positive and negative judgments for each identified topic. Retention of the multifaceted views is important for explorative case studies and reflects the richness of the views people have when they experience a phenomenon. Overall, the results appeared relevant, plausible, and without systematic bias. This assessment was confirmed by the validation presented in Sect. 5.1 and by the informants who were invited to provide feedback on the study results.

Internal validity concerns the causal character of the relationship between the concepts that were studied. One of the biggest risks of our study was that the laboratory situation may have biased the study results. In particular, observers could not be considered to be developers with substantial experience in the kind of system that was discussed in the workshop. Also, none of them had a vested interest in the requirements discussed in the workshop and did not experience the workshop video as a part of a development project. Instead, they watched a video that involved the first author of this paper and their lecturer as requirements engineer. As a result, there was a great risk that observers were biased and that they tried to please, rather than being honest with their answers.

The influence of the laboratory situation was controlled by comparing the obtained results with the results obtained from an in-project validation of the study results with a real representative of the development team. The influence of the student–lecturer relationship was controlled by a pre-briefing and the study protocol. The observers were explicitly pointed at this potential threat to validity and instructed to provide honest answers without consideration for the lecturer. Also, the points awarded for the feedback incented rich, trustworthy feedback and discouraged biased, pleasing feedback. For the in-project validation, there was no power relationship between the researchers and the respondent. No potentially biasing relationship existed for any other person involved in the study.

The good correspondence of the study results with the results from the in-project validation confirmed the validity of the presented results. However, the comparison showed that knowledge of the software system and of the development project had some influence: That knowledge reduced problems of video understanding and increased need for detail for well-known features. Thus, the presented results are slightly pessimistic in comparison with the appreciation of a workshop video by real project members. Also, the workshop discussions should be adapted to the knowledge of the recipients of the video workshop.

The power relationship between the lecturer and observers did not have any observable influence. The study results showed critical appraisal of the lecturer's performance as requirements engineer. Not only positive, but also a substantial amount of negative comments as well as recommendations was received for how to improve requirements engineering in the video-recorded workshop. Also, the judgments and feedback were consistent with the results of the real-world in-project validation where no power relationship existed.

External validity concerns the extent the findings can be generalized and are of interest to people outside the investigated case. Case study research does not aim at statistical generalization from a statistically representative sample. Instead, the presented study tried to describe the perception of the phenomenon of a workshop video by a developer as rich as possible. For research, these rich descriptions are then a basis for formulating hypotheses that can be tested in large scale with statistical methods. For practice, these rich descriptions provide guidance for how to well implement the technique and recommendations for how to best benefit from it. The biggest threats to external validity are the transferability of the results obtained from the laboratory situation to real-world contexts and from the specific workshop video to cases of workshop videos.

To assess transferability from laboratory to real-world projects, we have performed the presented in-project validation. As elaborated in the discussion of internal validity, the laboratory situation did not affect the perception of the workshop video significantly. When comparing the study results with the use of workshop videos in a real-world project, the validation results pointed to reduced problems in the understanding of video contents and a need to adapt the video contents to the knowledge of the video receivers.

When transferring workshop videos to projects than the presented one, we expect that the performance of the requirements engineer, video crew, and the stakeholders will affect the appreciation of the video. A video that implements our reported recommendations will be perceived more positively. These results were a main deliverable of this study and should be used for future workshop videos.

The role-play approach used in the requirements workshop suited well the character of the discussed software system. The system was intended to support a non-trivial business process and involved a substantial amount of interaction with various types of users. Information systems and Internet-based applications such as those envisioned by the future Internet are typical examples of such systems. In contrast, systems that barely involve end-user interaction, for example many types of middleware, will hardly benefit from the chosen requirements workshop approach and will require adaptation of the requirements engineering methodology. Replication of the study with positive and negative cases should be used to further test the external validity of the presented results.

Reliability concerns the extent the operations of the study can be repeated with the same results. The key concerns for reliability in the presented study relate to the instructions given to observers, the tooling used for annotating the workshop video, the analysis used to answer the research questions, and the rigor of the research.

A written case study protocol, an online video watching and annotation tool, and the questionnaire shown in the appendix were used to instruct observers and guide them through the study. The protocol included detailed instructions on the steps to follow. These instruments helped to replicate the study with observers who joined the study at a later moment, such as the head designer for in-project validation of the study results. All observers had followed the same procedure and handed in their data in a homogeneous format.

The analysis was performed in an attempt to enable auditability of the research for potential inspection. A case study database was used to collect the empirical data and to store intermediate and final analysis results. Two of the co-authors independently analyzed the video annotations and thematically categorized the answers collected with the questionnaire. Inconsistencies were then resolved by seeking consensus. One researcher was involved in the case, thus was able to provide background and to help interpreting the empirical data. The other researcher ensured neutrality. The joint parallel work and occasional consensus meetings reduced bias by consciously bracketing out prior experiences and other assumptions. The storage of empirical data and of the intermediate and final thematic analysis results gives a chain of evidence that connects the presented results to the collected empirical data.

6 Discussion

6.1 Workshop videos for requirements communication

The main contribution of this paper is in the proposal, a description, and a laboratory evaluation of workshop videos for requirements communication. The paper has described how the video technique can be transferred from workplace analysis, human–computer interaction research, and computer-supported cooperative work to the communication of requirements from workshop to developers. Based on the laboratory evaluation, the paper reported strengths of the technique, problems that can be encountered, and recommendations on how these problems may be overcome.

As illustrated in Fig. 1, requirements are surfaced by discussing them with stakeholders in a workshop, recorded with commonly available equipment as a video, and then handed over to developers with little or no editing. The technique is broadly applicable because workshops are one of the most common requirements engineering techniques [68], good video recording equipment started to be commonly available with smart phones and digital cameras, and hardly any requirements workshop involves all developers who later develop or evolve the software [45].

In this paper, we have motivated the need for this approach. We also explained our motivation for selecting the least sophisticated—and the least expensive—form of video. This type of video (recording) is characterized by:

- *Focus*: Interaction of stakeholder rather than the software product or its interface.
- *Goal*: Documentation of requirements rather than prototyping a potential solution or detailed evaluation for improvement.
- *Scope*: We consider both requirements and the interaction of stakeholders relevant, not just the resulting requirements. In particular, processes and use cases extending beyond one stakeholder's responsibility or view are of interest. For stakeholders, their personality, priorities, and rationale related to their goals are relevant. They are, therefore, in scope.
- *Status*: We assume stakeholders have a good understanding of their own requirements, but there may be gaps or inconsistencies when confronted with the views of other stakeholders. New insights can emerge during the workshop, and stakeholders will learn more about other perspectives.
- *Noninvasive*: Stakeholders can forget or ignore the presence of a camera. There is no script, storyboard, or directions given to them by the film crew. The camera is a strictly passive observer.
- *Resources*: Effort and cost for video preparation, recording, and post-procession need to be as low as possible.

The approach complements or, if well implemented and consistent with the software development process, may replace the hand-off of a written specification for communicating requirements. In comparison with the specification of a requirements document as suggested by standards such as ISO/IEC/IEEE 29148, the capture and storage of a video are more efficient because it requires less time and effort to produce the requirements artifact that can be handed off to development. This efficiency is especially important in early project phases where requirements are still settling, changing, and evolving. Also, the creation and storage of a video does not need as advanced skills are needed to specify a requirements document.

In comparison with frequent meetings between stakeholders and developer as suggested by agile development approaches, e.g. [4], a workshop video is a lightweight approach to solidify the discussions [39] and make them available to people who did not have the opportunity to participate in the meetings. The proposed approach suggests the creation of a video as a by-product [58] of a requirements workshop. A workshop video can be characterized as a by-product because it does not interfere with the workshop, it requires only little preparation or post-processing, and it does not necessitate expensive equipment. According to the results of the presented evaluation, the solidification of a workshop with a video allows introducing project members to the context, stakeholders, and requirements after the workshop and allows reducing the need to approach stakeholders when development decisions are taken. Workshop videos thus enable asynchronous communication with developers, even if they are remote in time and space.

A workshop video, finally, provides developers with a much richer source for information about requirements than text-based or more formal specifications. Workshop videos show how the system will be used, how stakeholders behave, the rationales they use when coming up with decisions, and how certain they are about the decisions. Rich inputs for requirements workshop videos may thus be one means to capture the "whys" behind decisions that otherwise would not be captured [8]. Rich inputs are especially effective when captured from a workshop held in the place where the discussed system will be used [17] and supported by prototypes that approximate the system [36].

6.2 Laboratory evaluation

The second contribution of this paper is a laboratory evaluation of workshop videos. We have applied our approach with 18 students who had experience in software development. Each student became an observer of the workshop and evaluated the video individually from the perspective of a developer tasked with the implementation of the system discussed in the video. The laboratory evaluation was validated with an important representative of the real development project that implemented the system. The validation results were consistent with the results obtained with the students except for features that were well known to the recipient and for his better understanding of the jargons used in the workshop by the stakeholders.

In a first step, observers' spontaneous reactions to the video were analyzed. Observers reacted to requirements, problems of the requirements workshop, interventions of the requirements engineer, and problems of requirements understanding. There seemed to be a permanent flow of perceived requirements throughout the video. Occasional problems emerged that distracted observers or reduced their understanding of the requirements. Some of these problems were addressed with the moderation of the workshop or resolved otherwise. Many requirements led to follow-up questions about the requirements that were not solved with the video. This result is consistent with results from other studies, which suggested that requirements understanding requires feedback between stakeholders and developers [46].

In the second step, we asked observers to rate the workshop video, reflect about strengths and weaknesses, and recommend improvements after seeing the video. Overall, observers would use workshop videos for requirements communication. Observers were moderately satisfied with the workshop video they saw. The large majority perceived the video to be useful for communicating requirements and felt able to start developing the system that was discussed in the video. Just few had been against the approach with the rationale that they would have preferred a complete, well-structured requirements document. These results indicate that workshop videos work well in situations where information about requirements is welcome and should be complemented with requirements specifications in situations that require specifications of high quality. Regulations, e.g., [9], or developer feedback may indicate such need.

Some of the strengths, weaknesses, and recommendations that observers indicated appear obvious and are reported in the literature about workshop moderation and video research. Other feedback was not anticipated. The feedback related to the workshop showed that there is a relationship between stakeholder preparation for the workshop and the appreciation of the workshop video from the developers. The more stable the stakeholder opinions are during a workshop, the more can the workshop video be used for information and educational purposes, thus is appreciated by developers.

A workshop video should be well structured and contain the information that is needed by the recipients. The somewhat diverging opinions for features well known to the video observer and for features that appear to be novel indicates that the depth of the information needs to be adapted to the recipient. This moving target was already observed in earlier studies [23] and is consistent with our results. An approach to achieve such adaptation may be to record a first video that provides a structured introduction into the ideas, context, and use of a planned software product and to complement this first result with in-depth video-recorded discussions that target the specifics of selected features, interfaces, or viewpoints in detail. Developer feedback may be used to assess the value produced with such additional solidified information and the risk of omitting it [69]. Alternatively, details may be specified with documentation as suggested by our observers.

Finally, disturbances, moderation problems, and video recording problems should be avoided to deliver an effective and convincing video-watching experience. Summarizing the feedback and recommendations from our observers, workshop videos should be created with a workshop moderator and participants who are well prepared and with an experienced video crew that utilizes professional video and audio recording equipment. However, such a workshop video is likely to be costly and effort-intensive. As no requirements practice alone suffices [70], we propose to resolve this trade-off by utilizing a lightweight approach and, where needed, compensate the lack of quality with ad hoc solutions. However, if the need for a high-quality workshop video is important enough and there are sufficient resources available, the approach can be upgraded.

Still, many of the recommendations reported in Sect. 4.4 do not hurt the lightweight character of the approach and should be considered for creating workshop videos in practice. For example, an agenda should be created, a protocol for moderation be used, the requirements engineer, film crew (even if just one person), and workshop participants be briefed on how to act during the workshop, and developers be supported in finding and interpreting the workshop video with complementing documentation. Figure 9 summarizes these findings and updates the conceptual model first shown in Fig. 1 above. It describes the aspects that should be retained or changed in order to achieve good results with workshop videos.

Compared to Fig. 1, there are now some of the desired additional documents involved. The software engineer uses a written protocol for moderation and follows a written agenda. Other documentation can be produced before the workshop, e.g., a description of the prototypes, during the workshop, e.g., a process description created on a whiteboard, or after the workshop, e.g., an index of the scenes contained in the video or a glossary. The recommendations presented in Sect. 4.4 are assigned to the different roles within the improved variant as indicated in Fig. 9 with the call-outs. Note that the outer box labeled “Requirements Communication” is the activity between stakeholders and developers we are ultimately interested in. Our technique describes all the details within that box, and this paper discusses the improvements and recommendations derived from applying the technique in the case study. The outer “interface” of the box remains the same: stakeholders and developers communicating.

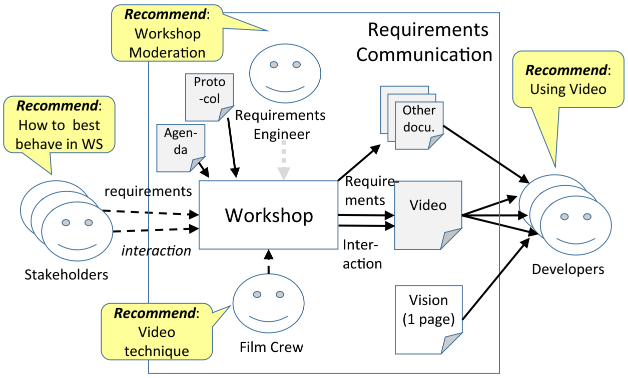

Fig. 9 Improved information flow, with recommendations

6.3 Future research

An evident step of future research is to further evaluate workshop videos as a technique for requirements communication. Dynamic validation of the presented approach in full-scale industry contexts is the next step to be performed for successful technology transfer [59]. Preferably, a longitudinal research approach should be chosen to show the effects of the technique on the emerging design and construction of the system. Thus, research should provide in-depth insights of what the technique means for software engineering, especially for the aspects of the workshop video technique that are considered to be negative according to the presented observer feedback. For example, automated extraction of a workshop protocol or indexing of the workshop video contents may help developers to find scenes relevant for answering their questions. Heath et al. [10] provide a good overview of such techniques that could be used to inform requirements engineering practice.

Even more important is to improve our understanding of asynchronous requirements communication. Workshop videos are one means to implement such communication. Another means is the use of a requirements specification. Questions that the presented study raise include: what information is essential to communicate to achieve good requirements understanding? In particular, what essential information do workshop videos communicate that specifications do not? For example, related research has shown that feedback from development to stakeholders is needed to ensure that developers truly understand the requirements [46]. According to these earlier results, hand-off of requirements, whether with video or with alternative means for documentation, would not lead to good-enough requirements understanding. However, the rich requirements information provided by workshop videos may give sufficient input to stimulate developer intuition and to allow developers to test assumptions and validating design alternatives before stakeholders are involved.

For successful implementation of requirements communication, whether with workshop videos or other technique, also the role of the participants needs to be investigated. The presented study raises questions such as:

How does the behavior and skills of the requirements engineer, the stakeholders, and the developers affect the outcome? How can the requirements receivers be sure they have correctly understood the requirements and give stakeholders the confidence that they were understood?

We expect that videos have a role to play in software engineering that is bigger than just requirements communication. To understand the potential and use of videos in the software development lifecycle, existing techniques need to be mapped and integrated and existing gaps identified. The workshop video technique can be combined with approaches defined by other researchers. Some are complementary to ours and could accommodate integration. For example, Creighton's Software Cinema technique describes how videos may be traced with UML diagrams [20]. Also, usability engineering may rely on videos for capturing screen content and facial expression of test users with tools such as Morae.[2] Such usability tests, used to evaluate implemented or approximated product features, may be defined in requirements workshops according to our approach. Further uses for video may include documentation of architecture and code, documentation of a planned or implemented product from the perspective of its users and for marketing and sales.

Videos may also start playing a role in empirical software engineering research. In the presented study, we have used the workshop video as a source for empirical data. We used that data for triangulation [66] of questionnaire-based feedback. The tagging of video contents and the convolution of the tags gave rich insights into what happened during the requirements workshop. These results show that video may be used to obtain rich insights about software engineering phenomena. The little work that has been done with videos so far, however, points to a need for better understanding the use of this increasingly adopted medium as an empirical source in software engineering research. Areas with a tradition of video analysis, such as computer vision, may offer the necessary theories and tools.

Finally, videos may become a tool for requirements engineering education. One of the problems of requirements engineering education is that the complexity of real-world requirements engineering is difficult to convey into a classroom [71]. To address this problem, videos may provide rich and deep insights into real-world activities and allow students to apprehend requirements engineering practice and stimulate their reflective observation. These learning outcomes may then increase the realism of the conceptualizations and experimentations performed by the students, thus closing the circle of experiential learning [72]. Systematic use of videos for requirements engineering teaching and learning has not been explored yet and is still in future research.

7 Summary and conclusions

Requirements have usually been communicated through specification documents, or in direct communication between stakeholders and developers. However, there are many cases in which neither of the two is effective: they are either too time-consuming, indirect, and error-prone (specification), or they are difficult to arrange and elusive (direct communication).

Video has long been proposed for documentation in software engineering. Its application to requirements engineering has been limited so far: Several authors proposed sophisticated techniques involving video. Their applicability is limited by the resources available in practical projects.

We suggest video documenting a workshop that focuses on requirements. The workshop encourages interaction among stakeholders for making requirements explicit and for validating expectations across stakeholder. Video recording preserves requirements raised and also the interaction between stakeholders. Thus, it facilitates remote and asynchronous communication between stakeholders and developers. In particular, we propose using low-cost video *as a by-product* of a workshop. We assume (almost) every project has the resources to use that technique.

The fundamental concepts of our approach are easy to explain and easy to follow, which we consider an important advantage. In this paper, we present an in-depth evaluation of an application case. The video was recorded in a real-world project. Then, 18 observers experienced in software development watched the video. We analyzed and compared both their spontaneous feedbacks, and their reflection afterward. As a result, we gained a better understanding of the perceived pros and cons of the technique from the perspective of a developer. Several recommendations for all roles involved could be derived, thus improving the technique. It mostly depends on available resources how many recommendations can be taken into account. A professional developer helped us in validating these findings against his experience background.

Requirements communication will not completely rely on videos instead of specifications and other documents. Instead, we propose video recording of requirements workshops as yet another technique available to the responsible requirements engineer.

There are many open research questions: What is the ideal setup for a given amount of resources? How much benefit will a professional film crew and some video

[2] http://www.techsmith.com/morae.html.

editing create and at what price? We presented video recording of a requirements workshop without considering the process and environment this will happen in. Identifying a balance of this technique with classical approaches for optimal overall requirements communication will be highly rewarding.

We intend to address these research questions by applying variants of our technique in different project contexts and with a variety of observers. Our final goal, however, will again not be polished videos—but affordable and effective requirements collaboration.

Acknowledgments This work was partially supported by the European Commission (FP7 project FI-STAR, Grant agreement no. 604691) and by the German Federal Ministry of Education and Research (K3 project 13N13548).

Appendix: evaluation questions

Table 25 shows the questions that were used for evaluating the observer's perception of a workshop video. The questions were contained in a questionnaire that was administered to a respondent after he or she had studied a workshop video in detail.

The question QQ1 was used to understand prior knowledge of the respondent. If learning plays a role in how a workshop video is perceived, prior knowledge of the system discussed in the workshop may affect the answers of the respondent.

The questions QQ2–QQ5 were posed as closed-ended questions for rating the studied workshop video. The idea was to develop a comprehensive judgment of the practice through triangulation. To understand the ratings, all answers had to be complemented with a rationale. QQ2 asked for an all-over-the-board judgment of the workshop video and was used to answer RQ2.1. Such questioning with the opinion score scale is common in quality-of-experience evaluation, e.g., [73]. At the place of the investigator, it is the respondent that decides about the criteria to be used for the judgment. In our case, we identified the criteria used by the respondents by asking them to justify the answers with a rationale. QQ3 was used in earlier requirements communication research [46], thus provides an opportunity to compare the answers. QQ4 was proposed for strategic planning of quality [65] with the explicit notions of good enough and competitiveness. QQ3 and QQ4 were used to answer RQ1.2. QQ5 finally elicits the willingness of the respondents to actually use a video for requirements communication. RQ3.1.

The questions QQ6–QQ8 were posed to understand the strengths and weaknesses of the workshop video when used for requirements communication, thus answering RQ1.3 and RQ2.3. The respondent was asked to take the perspective of a potential developer when answering these questions. To avoid complication of the study, we did not require the respondents to actually develop the solution. Instead, they were asked to use their prior software development experiences to identify plausible strengths

Table 25 Evaluation questions

About the respondent

QQ1. Do you have experience in building applications like discussed in the workshop? [Yes/No]

Video rating

QQ2. Overall, how satisfied are you with the video? [Opinion Score Scale]

QQ3. Overall, how capable do you feel you are able to implement the discussed solution discussed in the video? [Opinion Score Scale]

QQ4. Overall, how do you judge the use of video recording for communicating requirements to developers? [Strategic Planning Scale]

QQ5. Being a potential developer of the discussed solution, would you use a video for requirements communication? [Yes/No]

Video contents

QQ6. From the eyes of a potential developer of the discussed solution: Which parts of the video do you judge to be the most valuable inputs for development?

QQ7. From the eyes of a potential developer of the discussed solution: Which parts do you judge not to be useful as inputs for development?

QQ8. From the eyes of a potential developer of the discussed solution: Which parts were missing in the video and should have been covered as inputs for development?

Video use

QQ9. From the eyes of a potential developer of the discussed solution: How would you use the video to support your development work?

Recommendations

QQ10. From the eyes of a potential developer of the discussed solution: What should the requirements engineer do differently next time?

QQ11. From the eyes of a potential developer of the discussed solution: What should the filming crew do differently next time?

QQ12. From the eyes of a potential developer of the discussed solution: What documentation would you expect in addition to the video?

Conclusion

QQ13. Any other comment

and weaknesses. Thus, the answers reflect the reaction of a developer before he or she starts using the video for implementation. The split of the answers into data for RQ1.3 and RQ2.3 was made through content analysis. In the validation of the workshop video with the head designer from the real project, the opinions were based on actual development experience, thus reflecting the reaction of a developer during implementation. Both situations are relevant for the evaluation of workshop videos. The discussion of the validation with the head designer shows the similarities and differences in the two situations.

The questions QQ9–QQ12 were used to answer RQ3 by asking how the respondent would use the video to support development and by asking for recommendations for the improvement of requirements engineering, video filming, and accompanying documentation. Again, the respondents were basing their answers by assuming they may be starting development. The head designer was giving feedback from within development. The discussion of the validation with the head designer shows the similarities and differences of the two situations.

References

1. Glinz M, Wieringa R (2007) Stakeholders in requirements engineering. IEEE Softw 24:18–20
2. Glinz M, Fricker S (2014) On shared understanding in software engineering: an essay. Comput Sci Res Dev
3. Fricker S (2009) Pragmatic requirements communication: the handshaking approach: Shaker
4. Schwaber K (1995) scrum development process. In: Presented at the OOPSLA'95 business object design and implementation workshop, Austin, Texas
5. Gottesdiener E (2002) Requirements by collaboration: workshops for defining needs. Addison-Wesley Professional, Boston
6. Voinov A, Bousquet F (2010) Modelling with stakeholders. Environ Model Softw 25:1268–1281
7. Persson A (2008) The practice of participatory enterprise modelling: a competency perspective. In: Johannesson P, Söderström E (eds) Information systems engineering: from data analysis to process networks. IGI Publishing, Hershey
8. Dutoit A, McCall R, Mistrík I, Paech B (2006) Rationale management in software engineering. Springer, Berlin
9. World Trade Organization (2011) Understanding the WTO. World Trade Organization, Geneva, Switzerland
10. Heath C, Hindmarsh J, Luff P (2010) Video in qualitative research. Sage Publications, Thousand Oaks
11. Jirotka M, Luff P (2006) Supporting requirements with video-based analysis. IEEE Softw 23:42–44
12. Brill O, Schneider K, Knauss E (2010) Video vs. use cases: can videos capture more requirements under time pressure? Presented at the requirements engineering: foundation for software quality (REFSQ 2010), Essen, Germany
13. Broll G, Hussmann H, Rukzio E, Wimmer R (2007) Using video clips to support requirements elicitation in focus groups: an experience report. In: Presented at the SE 2007 workshop on multimedia requirements engineering, Hamburg, Germany
14. Pham R, Meyer S, Kitzmann I, Schneider K (2012) Interactive multimedia storyboard for facilitating stakeholder interaction. In: Presented at the 8th international conference of information and communication technology (Quatic 2012), Lisbon, Portugal
15. Zachos K, Maiden N, Tosar A (2005) Rich-media scenarios for discovering requirements. IEEE Softw 22:89–97
16. Rabiser R, Seyff N, Grünbacher P, Maiden N (2006) Capturing multimedia requirements descriptions with mobile re tools. In: Presented at the 1st international workshop on multimedia requirements engineering, Minneapolis, MN, USA
17. Maiden N, Otojare O, Seyff N, Grünbacher P, Mitteregger K (2007) Determining stakeholder needs in the workplace: how mobile technologies can help. IEEE Softw 24:46–52
18. Bevan N (1995) measuring usability as quality of use. Softw Qual J 4:115–130
19. Holzinger A (2004) Rapid prototyping for a virtual medical campus interface. IEEE Softw 21:92–99
20. Creighton O, Ott M, Bruegge B (2006) Software Cinema—Video-based Requirements Engineering. In: Presented at the 14th IEEE international requirements engineering conference (RE'06). Minneapolis, Minnesota, USA
21. Ivarsson M, Gorschek T (2009) Technology transfer decision support in requirements engineering research: a systematic review of REj. Requirements Eng 14:155–175
22. Fricker S, Grünbacher P (2008) Negotiation Constellations - Method Selection Framework for Requirements Negotiation. In: Presented at the requirements engineering: foundation for software quality (REFSQ 2008). France, Montpellier
23. Fricker S, Gorschek T, Byman C, Schmidle A (2010) Handshaking with implementation proposals: negotiating requirements understanding. IEEE Softw 27:72–80
24. Maiden N, Ncube C, Kamali S, Seyff N, Grünbacher P (2007) Exploring scenario forms and ways of use to discover requirements on airports that minimize environmental impact. In: Presented at the 15th IEEE international requirements engineering conference (RE'07). New Delhi, India
25. Gemino A, Wand Y (2005) Complexity and clarity in conceptual modeling: comparison of mandatory and optional properties. Data Knowl Eng 55:301–326
26. Lauesen S (2002) Software requirements: styles and techniques. Pearson Education, New York
27. Guarino N, Oberle D, Staab S (2009) What is an Ontology. In: Staab S, Studer R (eds) Handbook on ontologies. Springer, Berlin
28. Schneider K (2009) Experience and knowledge management in software engineering. Springer, Berlin
29. Wood J, Silver D (1995) Joint application development. Wiley, Hoboken
30. Gottesdiener E (2004) Running a use case/scenario workshop. In: Alexander I, Maiden N (eds) Scenarios, stories, use cases: through the systems development life-cycle. Wiley, Hoboken
31. Stirna J, Persson A, Sandkuhl K (2007) Participative enterprise modeling. Experiences and recommendations. In: Presented at the 19th international conference on advanced information systems engineering (CAiSE 2007), Trondheim, Norway
32. Sutcliffe A, Maiden N, Minocha S, Manuel D (1998) Supporting scenario-based requirements engineering. IEEE Trans Software Eng 24:1072–1088
33. Weidenhaupt K, Pohl K, Jarke M, Haumer P (1998) Senarios in system development: current practice. IEEE Softw 15:34–45
34. Seyff N, Maiden N, Karlsen K, Lockerbie J, Grünbacher P, Graf F et al (2009) Exploring how to use scenarios to discover requirements. Requirements Eng 14:91–111
35. Atladottir G, Hvannberg ET, Gunnarsdottir S (2012) Comparing task practicing and prototype fidelities when applying scenario acting to elicit requirements. Requirements Eng 17:157–170

36. Rettig M (1994) Prototyping for tiny fingers. Commun ACM 37:21–27
37. Ishida T, Hattori H (2009) Participatory technologies for designing ambient intelligence systems. Journal of Ambient Intelligence and Smart Environments 1:43–49
38. Stapel K, Schneider K, Lübke D, Flohr T (2007) Improving an industrial reference process by information flow analysis: a case study. In: Presented at the product-focused software process improvement (PROFES 2007). Riga, Latvia
39. Stapel K, Schneider K (2012) Managing knowledge on communication and information flow in global software projects. Expert systems—special issue on knowledge engineering in global software development
40. Stapel K, Knauss E, Schneider K (2009) Using flow to improve communication of requirements in globally distributed software projects. In: Presented at the collaboration and intercultural issues on requirements: communication, understanding and soft skills (CIRCUS 2009). Atlanta, Georgia, USA
41. Kwan I, Damian D, Storey M-A (2006) Visualizing a requirements-centred social network to maintain awareness wtihin development teams. In: Presented at the 1st international workshop on requirements engineering visualization (REV 2006). DC, USA, Washington
42. Gorschek T, Wohlin C (2006) Requirements abstraction model. Requirements Eng 11:79–101
43. Kruchten P (1996) A rational development process. CrossTalk 9:11–16
44. Royce W (1970) Managing the development of large software systems. In: Presented at the IEEE WESCON. Los Angeles, USA
45. Raijlich VT, Bennet KH (2000) A staged model for the software life cycle. Computer 33:66–71
46. Fricker S, Glinz M (2010) Comparison of requirements hand-off, analysis, and negotiation: case study. In: Presented at the 18th ieee international requirements engineering conference (RE'10). Australia, Sydney
47. Carter LR, Karatsolis A (2009) Lessons from trying to develop a robust documentation exemplar. In: Presented at the 27th ACM international conference on design of communication, Bloomington, Indiana, USA
48. Owen R, Stilgoe J, Macnaghten P, Gorman M, Fisher E, Guston D (2013) A framework for responsible innovation. In: Owen R, Bessant J, Heintz M (eds) Responsible innovation: managing the responsible emergence of science and innovation in society. Wiley, Hoboken, pp 27–50
49. Silas M, Grassia P, Langerman A (2015) Video recording of the operating room—is anonymity possible? J Surg Res
50. Bunn A (2015) The curious case of the right to be forgotten. Comput Law Secur Rev 31:336–350
51. Feeney W (1983) Documenting software using video. In: Presented at the IEEE computer society workshop on software engineering technology transfer. Miami Beach, FL, USA
52. DeMarco T, Geertgens C (1990) Use of video for program documentation. In: Presented at the 12th international conference on software engineering (ICSE 1990), Nice, France
53. Pham R, Holzmann H, Schneider K, Brüggermann C (2012) Beyond plain video-recording of gui-tests—linking test case instructions with visual response documentation. In: Presented at the 7th IEEE/ACM international workshop on automation of software test (AST 2012). Zurich, Switzerland
54. Alexander I, Maiden N (2005) Scenarios, stories, use cases: through the systems development life-cycle. Wiley, Hoboken
55. Schneider K, Stapel K, Knauss E (2008) Beyond documents: visualizing informal communication proceedings of third international workshop on requirements engineering visualization (REV '08)
56. Wieringa R, Maiden N, Mead N, Rolland C (2006) Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. Requir Eng 11:102–107
57. Miles M, Huberman M (1994) Qualitative data analysis: an expanded sourcebok. SAGE Publications, California, USA
58. Schneider K (2006) Rationale as a By-Product. In: Dutoit A, McCall R, Mistrik I, Paech B (eds) Rationale management in software engineering. Springer, Berlin, pp 91–109
59. Gorschek T, Wohlin C, Carre P, Larsson S (2006) A model for technology transfer in practice. Softw IEEE 23:88–95
60. Höst M, Regnell B, Wohlin C (2000) Using students as subjects: a comparative study of students and professionals in lead-time impact assessment. Empir Softw Eng 5:201–214
61. Ericsson KA, Simon HA (1980) Verbal reports as data. Psychol Rev 87:215–251
62. Horova I, Kolacek J, Zelinka J (2012) Kernel smoothing in matlab: theory and practice of kernel smoothing. World Scientific Pub Co, Singapore
63. Trochim W, Donnelly J (2008) The research methods knowledge base, (3rd edn) Atomic Dog
64. Hsieh HF, Shannon SE (2005) Three approaches to qualitative content analysis. Qual Health Res 15:1277–1288
65. Regnell B, Berntsson Svensson R, Olsson S (2008) Supporting roadmapping of quality requirements. IEEE Softw 25:42–47
66. Yin RK (2008) Case study research: design and methods. SAGE Publications, Thousand Oaks
67. Runeson P, Höst M (2009) Guidelines for conducting and reporting case study research in software engineering. Empir Softw Eng 14:131–164
68. Fricker S, Grau R, Zwingli A (2014) Requirements engineering: best practice. In: Fricker S, Thuemmler C, Gavras A (eds) Requirements engineering for digital health. Springer, Berlin
69. Glinz M (2008) A risk-based, value-oriented approach to quality requirements. IEEE Softw 25:34–41
70. Davis A, Zowghi D (2006) Good requirements practices are neither necessary nor sufficient. Requir Eng 11:1–3
71. Regev G, Gause D, Wegmann A (2008) Requirements engineering education in the 21st Century, an experiential learning approach. In: Presented at the 16th iEEE international requirements engineering conference (RE'08), Barcelona, Spain
72. Kolb D (1984) Experiential learning. Prentice Hall, Upper Saddle River, NJ, USA
73. Fiedler M, Hossfeld T, Phuoc T-G (2010) A generic quantitative relationship between quality of experience and quality of service. IEEE Netw 24:36–41