Ensuring Software Quality through Videos in Requirements Engineering

Jianwei Shi

Leibniz Universität Hannover, Software Engineering, Hannover, Germany

Abstract

Complex software systems are drafted, specified, and implemented in cooperation of all stakeholders. A shared understanding of requirements is important in all phases. However, requirements engineers, customers, developers, testers, and other stakeholders do not have a shared understanding of requirements due to insufficient communication. To solve this problem, this paper proposes to use videos to facilitate stakeholder involvement in requirements engineering. The basic idea is to use vision videos and test videos in requirements engineering with the help of testing techniques. A vision video visualises a future software system with a motivation and an envisioned solution. Requirement engineers use vision videos to elicit testable requirements from stakeholders and then validate the elicited requirements. A test video shows test executions of a software under development. Developers use test videos for (1) clarification of misunderstood requirements among development team and (2) eliciting new requirements and collecting feedback from stakeholders. Experiments and case studies are planned to check the effectivity or efficiency of these methods. The testability of a software is shaped at the beginning through defining testable requirements. During development, these requirements are updated through stakeholders' feedback. Hence, using video in requirements engineering ensures software quality.

Keywords

Video, test, requirements, feedback, software quality

1. Introduction

A successful working software should fulfil complex requirements from different stakeholders. Creating and maintaining a shared understanding of requirements among these stakeholders is challenging. The first challenge is before the software development. Requirements engineers are often not familiar with a domain, where customers come from. Customers may use domain definitions which requirements engineers misunderstand. Customers usually have little knowledge about the software technologies. Requirements engineers may talk about software technologies which customers do not understand. Because of this symmetry of ignorance [1], requirements engineers and customers cannot have the same understanding of requirements. The second challenge is during the software development, Spillner et al. [2] point out that

In: A. Ferrari, B. Penzenstadler, I. Hadar, S. Oyedeji, S. Abualhaija, A. Vogelsang, G. Deshpande, A. Rachmann, J. Gulden, A. Wohlgemuth, A. Hess, S. Fricker, R. Guizzardi, J. Horkoff, A. Perini, A. Susi, O. Karras, A. Moreira, F. Dalpiaz, P. Spoletini, D. Amyot. Joint Proceedings of REFSQ-2023 Workshops, Doctoral Symposium, Posters & Tools Track, and Journal Early Feedback Track. Co-located with REFSQ 2023. Barcelona, Catalunya, Spain, April 17, 2023.

jianwei.shi@inf.uni-hannover.de (J. Shi)

https://www.pi.uni-hannover.de/en/se/team/jianwei-shi (J. Shi)

0000-0001-6228-2478 (J. Shi)

CEUR Workshop Proceedings (CEUR-WS.org)

developers, testers, and users have different opinions on categorising a software behaviour into a bug or a feature. Studies [3, 4] in the recent decades show the missing communication between development and test teams. Testers cannot understand the expected behaviours because of missing involvement in defining requirements specification [5].

The purpose of this work is to use video to shape shared understanding of requirements among stakeholders. To address the first challenge, vision videos have been proposed to visualise a future software. This video contains a motivation and a visionary solution for the software. Karras et al. [6] have proposed production guidelines of vision videos. Instead of focusing on creation of vision videos, this work proposes that requirements engineers **use** a vision video to elicit and validate requirements. To address the second challenge, this research explores to use test videos from Graphical User Interface (GUI) tests for communication about software behaviours. This work applies the usage of test videos for developments where tests are available in the early stage. This work use existing tests to create test videos. A development team shows the test video in the team or with other stakeholders. Requirements are reviewed and communicated by viewing the video. Moreover, this work explores tracing between a vision video and a test video. The development team checks (1) if all envisioned requirements from a vision video are implemented or considered in a test video and (2) if each implemented functionality is consistent with the vision.

This work is organised as follows: Section 2 lists related work. Research questions and methods are explained in Section 3. Achievements so far and further research plan are presented in Section 4. Section 5 concludes this research.

2. Related Work

In Requirements Engineering (RE). videos are created and used which can visualise visionary scenarios of a future software [7, 8]. To create a model for a software that is not yet implemented, Creighton et al. [7] propose a video-based methodology for requirements development. In an experiment, Brill et al. [8] explore to create videos and compare videos with use cases. The results show that customers tend to confirm or correct more basic requirements (in Kano model) with videos than use cases. For creating vision videos, Karras and Schneider [6] propose guidelines. Instead of video creation, this work focuses on video usage.

Videos are also used in capturing GUI interactions. For debugging, Pham et al. [9] record GUI tests and trace video to test code. By using the video viewer in a company, two developers mention that the debugging is faster. Shi and Schneider [10] follow this concept and propose to highlight GUI interactions. Similarly, Karras et al. [11] use videos to record GUI interactions based on mock ups. They claim that the video can help developers understand system usage better than static mock ups. This work proposes to use test video not only for testers and developers, but also for end-users, marketing people, and other stakeholders to communicate about requirements.

Requirements should fulfil quality-related criteria to be testable. Generally, Robertson [12] sets the quality gateway to ensure accurate requirements which are for clear communication in design and implementation. Concretely, Robertson and Robertson [13] suggest to use an requirement shell to write an requirement. The “Fit Criteria” field in the requirement shell

defines how to test the requirement, i.e. how a system should behave. This work follows this principle and considers testability of requirements in elicitation and validation by using vision videos.

Involving test activities into RE is mentioned in many works [14, 15, 16]. Uusitalo et al. [14] investigate challenges and benefits of the practices in an industrial interview and a literature review. They argue that implicit information is made clear and accessible by defining requirements with testers. Bjarnason et al. [15] suggest that developers and testers define testable requirements together. In addition, they find the early user testing is important because feedback can be collected early and considered in development. Unterkalmsteiner et al. [16] define the alignment of RE and software testing (ST) as "the adjustment of RE and ST efforts for coordinated functioning and optimised product development". They underline the need for alignment and propose a taxonomy to classify methods of alignment. This work concentrates on using test practices in RE and proposes concrete video usage methods.

3. Research Method

This work aims to use videos for facilitating communication about requirements. Figure 1 shows RE activities where videos can be used. Related Research Questions (RQ) are marked in Fig. 1 and will be explained in this section.

Firstly, this work proposes to use vision videos for elicitation and validation of testable requirements. In this work, testable requirements should be understandable, accurate and complete for communication. This paper proposes to use a template of acceptance criteria to formulate testable requirements, while viewing a vision video. The requirements are written in natural language and contain elements of a test case: preconditions, actions with inputs, and expected results. The requirement format is chosen for the following reasons: (1) stakeholders can understand the requirements and give feedback; (2) accurate descriptions can be specified in inputs of the actions and (3) complete requirements are ensured by specifying all test case elements. This work asks:

- **RQ1**: How to use vision videos to elicit testable requirements effectively?
- **RQ2**: How to use vision videos to validate the elicited testable requirements efficiently?

Effectiveness in RQ1 is defined as elicitation of accurate and complete requirements. Efficiency in RQ2 is defined as validation in short time and with necessary minimal communication effort. For RQ1 and RQ2, this work focuses on functional and quality requirements. We want to conduct empirical experiments to answer these questions.

Secondly, this work suggests to use the test video to communicate about requirements during development. Test videos are generated while executing GUI tests. These tests can be written from testable requirements, which are elicited and validated through vision videos. GUI tests are also available at the early stage of a development if a team chooses the test-driven development (TDD) or applies TDD during development (e.g. behaviour-driven development, BDD). This research will create test videos by executing existing GUI tests, and use the test video to help testers and developers understand requirements. Among development teams, testers and developers communicate about requirements with test videos. Additionally, the

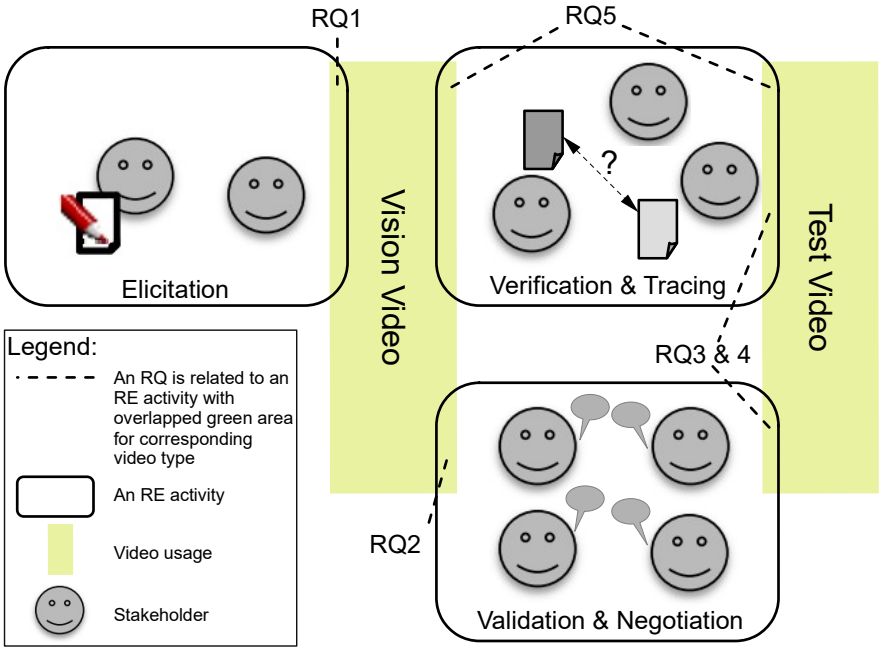

Figure 1: The use of vision video and test video in RE. (Based on Fig. 1 from Brill et al. [8]) Requirement engineers use a vision video for eliciting requirements from stakeholders (RQ1). Requirement engineers use a vision video to validate the elicited requirements (RQ2). A test video is created during a GUI test execution. Testers and developers communicate about requirements by using the test video in verification (RQ3 & 4). In validation and negotiation, stakeholders can see the test video and give their feedback (RQ3). Lastly, the vision video and the test video are used together for tracing (RQ5).

test video serves as a demonstration of the software under development. Stakeholders can give feedback on that, thus requirements can be updated during development. This paper asks:

- **RQ3**: How to use test videos to activate communication between stakeholders for a shared understanding of requirements?

Experiments will check how a test video helps different stakeholders, concretely following evaluation questions (EQ):

- **EQ1**: How do test videos help testers find a defect?
- **EQ2**: How do test videos help developers understand a reported defect?
- **EQ3**: How do test videos help stakeholders find inconsistencies in implementation?
- **EQ4**: How do test videos help stakeholders find missing requirements?

For EQ1 and EQ2, a test video shows a failed GUI test. For EQ1, testers review the video for debugging. For EQ2, developers receive a test video as a defect report from testers. Two reasons are possible for the test failure: (1) the test failure reveals a real defect (error in source code); (2) testers misunderstand requirements and write incorrect specified test code. Hence, understanding defect in EQ2 means that a developer can differentiate between error in source code and incorrect specified test code. For EQ3 and EQ4, a test video shows a successful GUI

test. EQ3 and EQ4 focus on stakeholders who are not directly involved in development, such as customers, end-users, and designers.

Thirdly, this work explores to use vision videos and test videos for the alignment of RE with ST, i.e., use test practices in RE. Based on RQ3, this work focuses on testing in agile or hybrid software development, e.g., TDD, BDD. Based on test activities in this development, we propose to conduct video generation when successful GUI tests are available.

For vision videos, answers of RQ1 and RQ2 already apply test practices in RE. For test videos, this work asks:

- **RQ4**: How to use test practices in RE with the help of a test video?

Besides regular development steps, this research embeds video creation and application steps in-between. Stakeholders who participate in communication by using the test video will be defined, e.g., *developers* show *end-users* the test video and ask for feedback from *end-users*.

Lastly, if a vision video and a test video are used in a software project, it is meaningful to investigate:

- **RQ5**: How to use a vision video and a test video together for tracing requirements?

For RQ4 and RQ5, case studies in industrial or academical settings are planned. Subjective opinions will be collected in interviews and coded. Quantitative metrics can be collected according to the Goal Question Metric method (GQM) [17].

4. Proposed Solution and Plan

For viewing vision video, a web tool *ViViPlayer* has been developed in a student software project. ViViPlayer is used to replay a vision video in segmented shots and to collect requirements and feedback for each shot. In a current preliminary study, participants view vision videos in ViViPlayer and write acceptance criteria. A semi-structured natural language *Gherkin* is used to formulate acceptance criteria, i.e. testable requirements. We assume that the structure of Gherkin could help make the requirements explicit. Reason of the assumption is: Although a vision video shows the requirements, they could be implicit presented and not be elicited. Hence, the structure of Gherkin could help make the requirements explicit. According to feedback of this preliminary study, we will enhance ViViPlayer and design a new experiment to answer RQ1 and RQ2.

To create and replay a test video which is linked to test code, another tool suite *ScreenTracer* is further implemented and extended [9, 10, 18]. EQ1 is checked in a preliminary experiment among testers in a company [18, 19]. Coding results [19] show that the highlight function helps reveal position of defects accurately and explicitly. Another experiment among developers is planned for checking EQ2.

Shi and Schneider [20] have presented a concept of creation and usage of a test video in development in an RE section meeting of German Computer Science Society. Mönnich [21] has concretised the concept and conducted an online experiment to check EQ3. Results show that test videos can help participants recognise inconsistencies between given requirements and demonstrated functionalities in the videos easily. For EQ4, this work plans to use requirement

documentations (e.g. specifications, user stories) with a test video to help stakeholders find missing requirements. RQ3 will be answered based on answers of all evaluation questions.

A test video is demonstrated to stakeholders for giving feedback about implemented software or generating new requirements. Stakeholders who are not directly involved in development, e.g., user experience designer, end-user, marketing personal, can give valuable feedback. Mönnich [21] presents how this concept applies in BDD. Further adaptations of the concept and the ScreenTracer are planned for the real project setting. A survey can be conducted among practitioners to investigate what benefits and constraints the approaches of using test video can bring and what success criteria should be. A case study will be conducted to evaluate the approaches in the perspective of development teams according to the survey result. A complete concept and results of the case study will answer RQ4.

For RQ5, a literature research is planned to check existing practices of tracing requirements between textual specifications and tests. These practices will be analysed for applicability in a vision video and a test video. Specifications can be extracted from a vision video, while test code is linked with a test video. A preliminary study is planned to evaluate applied practices in a student software project or laboratory. The study will check (1) which requirements shown as visionary scenarios in the vision video are implemented, changed, or ignored; and (2) if all implemented functionalities in the test video are in accordance with the project vision shown in the vision video. A semi-automatic method of tracing will be explored to conduct these checks.

In summary, this research has following next steps: Experiment design, conduction and further steps for (1) **RQ1** and **RQ2**, (2) **EQ2**, (3) **EQ4**; and (4) Conception, survey, and case study for **RQ4**; (5) Literature research and preliminary study for **RQ5**.

The proposed solution has following contributions to existing literature: (1) Concrete usage methods of vision videos and test videos are proposed in elicitation, verification, validation, and negotiation in RE; (2) Software quality is ensured by defining testable requirements and communicating about them; (3) Dedicated user studies are planned to check the feasibility of proposed concepts in real project settings.

A challenge of the plan would be the difficulty finding a company, if an industrial case study will be conducted to answer RQ4. Due to privacy issues, companies may not willing to share their requirements with researchers. This challenge should be overcome by active communication in conferences for possible contacts. Another challenge is the tight time budget (i.e. approximately two years) for the whole research. This challenge should be solved by (1) regular report with supervisor and (2) cooperation with students and partners.

5. Conclusion

This research uses video as a communication medium in RE to share the same understanding of requirements between stakeholders. Vision videos are used in RE to activate communication and the usage makes tacit knowledge explicit (i.e. as testable requirements) for all stakeholders. Test videos are created from test executions and used to (1) align the understanding of requirements among developers and testers and (2) involve stakeholders in development. The tracing between a vision video and a test video can help a project leader inform the current project status and make a decision. A good software quality is ensured before and during development. Vision

videos and test videos help shape a shared understanding of requirements and maintain a good software quality throughout development.

Acknowledgments

This work is partly funded by Deutsche Forschungsgemeinschaft (DFG) - Project number 289386339 (ViViUse).

References

[1] G. Fischer, Symmetry of ignorance, social creativity, and meta-design, Knowledge-Based Systems 13 (1999) 527–537. doi:10.1016/S0950-7051(00)00065-4.

[2] A. Spillner, T. Roßner, M. Winter, T. Linz, Praxiswissen Softwaretest - Testmanagement: Aus- und Weiterbildung zum Certified Tester - Advanced Level nach ISTQB-Standard, dpunkt, Heidelberg, 2014.

[3] E. Bjarnason, K. Wnuk, B. Regnell, Requirements are slipping through the gaps - a case study on causes & effects of communication gaps in large-scale software development, in: 2011 IEEE 19th International Requirements Engineering Conference, IEEE, Trento, Italy, 2011, p. 37–46. doi:10.1109/RE.2011.6051639.

[4] X. Zhang, T. F. Stafford, J. S. Dhaliwal, M. L. Gillenson, G. Moeller, Sources of conflict between developers and testers in software development, Information & Management 51 (2014) 13–26. doi:10.1016/j.im.2013.09.006.

[5] D. S. Cruzes, N. B. Moe, T. Dyba, Communication between developers and testers in distributed continuous agile testing, in: 2016 IEEE 11th International Conference on Global Software Engineering (ICGSE), IEEE, Orange County, CA, USA, 2016, p. 59–68. doi:10.1109/ICGSE.2016.27.

[6] O. Karras, K. Schneider, An Interdisciplinary Guideline for the Production of Videos and Vision Videos by Software Professionals, 2021. URL: https://arxiv.org/abs/2001.06675.

[7] O. Creighton, M. Ott, B. Bruegge, Software cinema-video-based requirements engineering, in: 14th IEEE International Requirements Engineering Conference (RE'06), IEEE, Minneapolis/St. Paul, MN, 2006, p. 109–118. doi:10.1109/RE.2006.59.

[8] O. Brill, K. Schneider, E. Knauss, Videos vs. use cases: Can videos capture more requirements under time pressure?, in: Requirements Engineering: Foundation for Software Quality, volume 6182, Springer Berlin Heidelberg, Essen, 2010, p. 30–44. doi:10.1007/978-3-642-14192-8_5.

[9] R. Pham, H. Holzmann, K. Schneider, C. Brüggemann, Tailoring video recording to support efficient gui testing and debugging, Software Quality Journal 22 (2014) 273–292. doi:10.1007/s11219-013-9206-2.

[10] J. Shi, K. Schneider, Creation of human-friendly videos for debugging automated gui-tests, in: Testing Software and Systems, Springer International Publishing, Cham, 2021, p. 141–147. doi:10.1007/978-3-031-04673-5_11.

[11] O. Karras, C. Unger-Windeler, L. Glauer, K. Schneider, Video as a by-product of digital prototyping: Capturing the dynamic aspect of interaction, in: 2017 IEEE 25th International

Requirements Engineering Conference Workshops (REW), IEEE, Lisbon, Portugal, 2017, p. 118–124. doi:10.1109/REW.2017.16.

[12] S. Robertson, An early start to testing: How to test requirements, in: Proc. of the EuroSTAR conference, Amsterdam, 1996.

[13] S. Robertson, J. Robertson, Reliable requirements through the quality gateway, in: Proceedings. Tenth International Workshop on Database and Expert Systems Applications. DEXA 99, IEEE, Florence, Italy, 1999, p. 357–363. doi:10.1109/DEXA.1999.795193.

[14] E. J. Uusitalo, M. Komssi, M. Kauppinen, A. M. Davis, Linking requirements and testing in practice, in: 2008 16th IEEE International Requirements Engineering Conference, IEEE, Barcelona, Spain, 2008, p. 265–270. doi:10.1109/RE.2008.30.

[15] E. Bjarnason, P. Runeson, M. Borg, M. Unterkalmsteiner, E. Engström, B. Regnell, G. Sabaliauskaite, A. Loconsole, T. Gorschek, R. Feldt, Challenges and practices in aligning requirements with verification and validation: a case study of six companies, Empirical Software Engineering 19 (2014) 1809–1855. doi:10.1007/s10664-013-9263-y.

[16] M. Unterkalmsteiner, R. Feldt, T. Gorschek, A taxonomy for requirements engineering and software test alignment, ACM Transactions on Software Engineering and Methodology 23 (2014) 1–38. doi:10.1145/2523088.

[17] V. Basili, H. Rombach, The tame project: towards improvement-oriented software environments, IEEE Transactions on Software Engineering 14 (1988). doi:10.1109/32.6156.

[18] M. A. Tandun, Report-video production for quick bug-finding in the web applications, 2022. URL: https://www.pi.uni-hannover.de/fileadmin/pi/se/Stud-Arbeiten/2022/BA_Tandun_2022.pdf, Bachelor Thesis under Jianwei Shi's Supervision.

[19] J. Shi, K. Schneider, M. Tandun, O. Karras, Supplementary material for evaluating screentracer among testers, 2023. doi:10.5281/zenodo.7522978.

[20] J. Shi, K. Schneider, Erstellung und Einsatz von Test-Videos für die Verständigung zwischen Kunden. Anforderungsingenieur, Tester und Entwickler, in: Softwaretechnik-Trends, volume 42 of *3*, 2022.

[21] J. Mönnich, Organisation von GUI-Unittests für frühe Rückmeldungen von Kunden, 2022. Master Thesis under Jianwei Shi's Supervision.