

QoE probe: A requirement-monitoring tool

Farnaz Fotrousi¹, Samuel A. Fricker^{1,2}

¹ Blekinge Institute of Technology, SERL-Sweden,
37 179 Karlskrona, Sweden
{farnaz.fotrousi, samuel.fricker}@bth.se
FHNW, i4Ds Centre for Requirements Engineering,
5210 Windisch, Switzerland
{samuel.fricker}@fhnw.ch

Abstract. Runtime requirement monitoring is used for verification and validation of implemented requirements. To monitor the requirements in runtime; we propose a “QoE probe” tool, a mobile application integrated through an API, to collect usage logs as well as users’ Quality of Experience (QoE) in the form of user feedback. The analysis of the collected data guides requirement monitoring of functional and non-functional requirements as well as capturing new requirements.

1 Motivation and Objective

Runtime requirement monitoring helps requirement engineers to verify the alignment of a software system with specified requirements [1] and to validate the users’ acceptance of the system’s requirements [2]. The monitoring facilitates understanding of requirements violation, i.e. the system fails to meet the specified requirements, discovering and prioritizing user needs through the running version of software system [3], and also facilitates a fast reaction for evolving the system and the corresponding requirements [4]. Understanding the requirement violation and real user needs at runtime requires an automated monitoring tool.

Available automated monitoring tools do not collect and align runtime logs and user feedback corresponding to different sets of requirements. Collecting just runtime logs gives insight into the implemented requirements, but does not reflect user satisfaction with the requirements or the system. User feedback alone cannot be traced back technical specifications of the system. The above discussion implies the importance of an automated user feedback collection tool that aligns user perception of the implemented requirements with the technical characteristics of the system.

We propose “QoE Probe” tool that captures users’ Quality of Experience (QoE) [5] by periodically requesting for user feedback, while collecting and aligning continuously with usage logs. The aligning feature and the provided API for flexible investigating of requirements make the QoE probe different than similar monitoring tools [6]. By analyzing the aligned usage logs and user feedback, “QoE probe” assists to find out whether users are satisfied with the implemented functional requirements, to discover new user requirements from the analysis of user feedback, and to find out the proper level of non-functional requirements based on the users’ QoE [7].

2 Tool Descriptions

“QoE probe” is an application for Android and iOS devices, integrated with a mobile application through an API. The API enables specifying the scope of features and user interactions, and the points that a user feedback should be collected. The log generated by “QoE probe” contains user-id, timestamps of requirement events in the feature level (e.g. starting or completing a feature) and user interaction level (i.e. a user input or an application output). In completion of a particular user interaction, feature or group of features, based on a defined likelihood, a user feedback form is triggered to collect the level of users’ acceptance and user comments. The records of user feedback together with the usage log-file are both stored on the users’ device and being sent to a server using a JSON protocol for analysis.

The “QoE probe” tool was used successfully in six use-cases of FI-STAR project as a sub-project of the European Future Internet Public-Private-Partnership (FI-PPP) program. Currently, the “QoE probe” is limited to the mobile application and does not include analysis components.

The latest version of “QoE probe” can be downloaded in Android [8] and iOS [9]. Using the “QoE probe” requires installation on a device, configuration of the probe, and integration with the target application. The integration happens through calling an API to tag the scopes of features or user interactions at the code level of the target mobile application. All steps will be presented during the tool session.

References

1. Wang, Y., McIlraith, S.A., Yu, Y., Mylopoulos, J.: An automated approach to monitoring and diagnosing requirements. Proceedings of the twenty-second IEEE/ACM international conference on Automated software engineering. ACM, Atlanta, Georgia, USA (2007)
2. Sampaio do Prado Leite, J.C., Freeman, P.A.: Requirements validation through viewpoint resolution. Software Engineering, IEEE Transactions on 17, 1253-1269 (1991)
3. Gomaa, H., Scott, D.B.: Prototyping as a tool in the specification of user requirements. 5th international conference on Software engineering. IEEE Press, San Diego, California, USA (1981)
4. Leucker, M., Schallhart, C.: A brief account of runtime verification. The Journal of Logic and Algebraic Programming 78, 293-303 (2009)
5. Le Callet, P., Möller, S., Perkis, A.: Qualinet white paper on definitions of quality of experience. European Network on Quality of Experience in Multimedia Systems and Services, (2012)
6. Froehlich, J., Chen, M.Y., Consolvo, S., Harrison, B., Landay, J.A.: MyExperience: a system for in situ tracing and capturing of user feedback on mobile phones. 5th international conference on Mobile systems, applications and services (MobiSys2007), San Juan, Puerto Rico (2007)
7. Fotrousi, F., Fricker, S.A., Fiedler, M.: Quality requirements elicitation based on inquiry of quality-impact relationships. 22nd IEEE International Conference on Requirements Engineering. IEEE, Karlskrona, Sweden (2014)
8. <https://github.com/farnazfotrousi/QoE-Probe-Android>
9. <https://github.com/LTFE/QoE-Probe-iOS>