

# Prioritizing Software Requirements by Combining the Usage Monitoring Data With Feedback Data

Syeda Synnia Tanveer<sup>1</sup>

<sup>1</sup> National University of Computer and Emerging Sciences, Lahore, Pakistan

## Abstract

Two of the fundamental sources of data for requirement gathering for system upgradation are usage monitoring data and feedback data. Work has been done to acquire both individually and collectively. This work aims at validating the existing methods for combining feedback and monitoring data on a large scale and taking it a step further by developing a recommender system for requirement elicitation and prioritization. We have developed prototypes for data gathering agents built in JavaScript and connected to our server through REST APIs so that we don't affect the original code base of deployed applications. Once done with testing locally we will take it to production and proceed further. Validation will be done on SAKAI learning management system implementations in educational institutions. The software requirements prioritization process will be evaluated before and after applying the recommender system. Causal-comparative/quasi experimental research method will be used to evaluate the system developed.

## Keywords

Software requirements elicitation, Requirements prioritization, Combining the Usage Monitoring Data with the Feedback Data, Correlation analysis of feedback and monitoring data, Recommender System for Software Requirement Prioritization

## 1. Introduction

Software evolves and adapts continually according to the user's ever-changing requirements. Requirements are elicited from the stakeholders continuously so that their needs concerning the particular software are met [1]. A record is maintained for violations of requirements [2], and a process to take action according to evolving requirements is incorporated [3]. Traditionally the elicitation process includes stakeholder interviews, user workshops, product prototyping. These processes can conveniently be managed in an organizational setting. However, they are challenging to manage in a modern context where a globally distributed audience is using the application. Frequent requirements elicitation and validation is the main challenge [4]. Existing methods of requirements engineering include crowd requirements engineering [12] and automatic classification of apps [13]. An efficient and scalable approach is needed to address this challenge of requirement elicitations from such a remote and diverse audience. Analytical reading of logs and application of data mining processes makes it convenient for the development teams to comprehend user feedback.

Monitoring and Feedback data collection frameworks are developed with a goal of continuous improvement in requirement elicitation. Such systems have components related to acquisition and storage of data. Further enhancement is made to these components by incorporating a methodology where monitoring data will further be aggregated using matrices along with runtime feedback in order to enable clustering and grouping of feedback statistics. This improves the quality of information

---

In: F.B. Aydemir, C. Gralha, S. Abualhaija, T. Breaux, M. Daneva, N. Ernst, A. Ferrari, X. Franch, S. Ghanavati, E. Groen, R. Guizzardi, J. Guo, A. Herrmann, J. Horkoff, P. Mennig, E. Paja, A. Perini, N. Seyff, A. Susi, A. Vogelsang (eds.): *Joint Proceedings of REFSQ-2021 Workshops, OpenRE, Posters and Tools Track, and Doctoral Symposium, Essen, Germany, 12-04-2021*

EMAIL: 1175507@lhr.nu.edu.pk



© 2020 Copyright for this paper by its authors.  
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).  
CEUR Workshop Proceedings (CEUR-WS.org)

presented to the requirements engineer, enhancing the elicitation process. Furthermore to enhance the prioritization process there is a need to develop a recommender system.

## 2. Problem

Frameworks developed for continuous requirements elicitation through combining both usage monitoring and user feedback have been tested on a limited data where data set was small and fragmented. We want to draw conclusion on quantitative benefits of using the above mentioned approach based on a sizable data set. Components for storing and combining data need to be developed, incorporating advance aggregation and clustering techniques. Based on the results gathered from this clustering and aggregations new requirements should be elicited and prioritized. There is a need for developing a recommender system that prioritizes requirements based on combination of user feedback and monitoring data.

## 3. Relevance

Literature review identifies that internal sources for user feedback, tools should be implemented to automate user feedback processing; third, user feedback to requirements should be related through reference points; requirements validation for user feedback should be enhanced along with establishing developer–user communication [14]. Data driven approach towards requirement engineering has also been presented [15].

Several researchers have advocated the usage of monitoring data alongside feedback data as a way of eliciting new requirements. As an example, in order to facilitate elicitation process there are approaches [5], [6] that use a combination of monitoring and feedback data like examining both user submitted forms and data gathered from logs. The challenge faced in this approach was that the data from those users who didn't submit a feedback could not be included. So it was hard to provide exact Number of users facing a particular issue.

In contrast, a method [7] used logs from entire user base and used data science methods to assess actual user behavior, resulting in more requirements. The authors purposed that monitoring information gathered through this process can be used in combination with feedback leading to a refined and well prioritized requirement set. The implementation of this idea was left for future work.

Few solutions exist that use monitoring and feedback data together for other purposes. QoE probe [8] is its example. It's a mobile app that capitalizes on monitoring and feedback data for verification and validation of requirements.

Probe requests feedback from time to time and continuously aligns it with system logs. Usage data is recorded with timestamps. It's recorded when a feature usage started and how long it ended and how the user interacted with the system. Whenever an interaction is completed user is presented with a dialogue box to provide feedback. This gives a thorough insight about user acceptance level and how much they resonance with user needs. This comprises of feedback and monitoring data but lacks analysis component [8].

MyExperience [9] is a similar tool combining monitoring and feedback data. This system facilitates studies regarding human health as health metrics are populated through IOT devices and directly took feedback about how they felt. The concepts presented in this work haven't been technically implemented.

FAME framework was presented, with an objective to facilitate requirement engineers in collecting and analyzing the monitoring and feedback data together with respect to a software system. User feedback is provided by end users using feedback modules while the real time events are monitored

through monitoring agents in the host application that are then sent to a data lake. Data from all sources is combined through ontology. The requirement engineer is presented the combined data used to elicit requirements [10]. In this work data pre-processing and incorporation of data storage and clustering and aggregation of feedback and monitoring data is left as future work. It also lacked grouping of feedback and monitoring data and calculating metrics rather than just recording run time events.

An effort has been made to let the system elicit user requirements on its own by using monitors that detect certain situations occurring while system use. On such occurrences a system generated request is sent. Feedback is then correlated to monitoring data at that instant. This was implemented in a smart city project and positive results were achieved by combing real time monitoring with preemptive feedback mechanisms. The techniques implemented showed enhancement in the requirement elicitation process and improved the end-user experience [11]. This paper lacked the statistical significance and the study was specific to IOT systems.

A systematic literature review done in 2020 concludes that more machine learning and repository-mining based research for requirement prioritization is needed, there is a need for addressing match between software project contextual settings and methods, it is required to evaluate and compare requirement prioritization methods in real-world [16].

The review of literature done above establishes the interest in eliciting requirements from feedback and monitoring data. Validating the frameworks presented in the literature on a sizable dataset and using advanced techniques for clustering and aggregation is a future work consideration [10] as indicated in the above mentioned literature.

A need for a requirement prioritization recommender system that combines user feedback and usage monitoring data is established from literature.

## **4. Novelty**

We will be testing the combined effect of utilizing both feedback and monitoring data onto a large real-time user base. We will be taking the data through tools that would not affect the original code base and will work as an extension. We would be developing a recommendation system that prioritizes software requirements gathered through the combination of user feedback and usage monitoring data.

## **5. Research Method**

A detailed literature review was done to establish the relevance of research. Search terms used were: user feedback data, usage monitoring, combination of usage monitoring and user feedback data, and recommendation system for requirement prioritization, upon literature review it was established that there is a need for a recommendation system that prioritizes software requirements gathered through the combination of user feedback and usage monitoring data. The validation has to be done in real-time production environment.

Following steps will be performed for developing the above mentioned system

Step 1: Shortlisting feedback and monitoring metrics

Step 2: Collection of feedback and monitoring data

Step 3: Preprocessing of feedback and monitoring data

Step 4: Clustering and aggregation on feedback and monitoring data

Step 5: Eliciting and prioritizing requirements based on the processed data

Step 6: Quantitative and qualitative comparison of elicitation process prior and after applying the framework

If step 6 is successful, we will perform correlation analysis that includes data reduction and correlation coefficient calculation. Then we will develop a recommender system using correlation based collaborative filtering and similarity score calculations. We should be getting elicited and prioritized requirements as an output. This framework will be validated on SAKAI learning management system implemented in universities.

By far we have shortlisted feedback and monitoring metrics. We have developed agents to monitor following metrics:

- Active users in a given period
- Sessions per active users
- Time spent on a feature
- Activity abandonment
- Activity completion
- Crash instances
- Bug reporting
- Feedback provision

**Initially these metrics will be applied to evaluate following modules of SAKAI**

- Announcements portal
- Assignments portal
- Academic and social alerts
- Calendar and scheduling
- Feedback Area
- Forums
- Gradebook Management
- Lessons Management
- Syllabus Management
- Tests & Quizzes

Causal-comparative/quasi experimental research method will be used to evaluate the system developed. We will evaluate the requirement prioritization process before and after implementation of the recommender system.

We will apply the recommender system to the SAKAI learning management system in three universities in Pakistan. The end users are thousands in number. We will collect data regarding software prioritization before and after implementation of the system from the system developers and we will analyze the data collected through the system. Usage data will be collected before and after taking feedback and monitoring data will be collected to access how usage was effected after prioritization through our recommender system.

By far we have developed an understanding around SAKAI architecture and deployment and we have reached a conclusion that we will be adding our agents to the front end of the application on through JavaScript code and will link this code to our server through REST APIs. We have developed the basis for collecting monitoring data. This will give us scalability and the code base of the production application won't be disturbed. We have developed a prototype of relevant JavaScript code and REST APIs and currently we are in process of integration with different modules of the SAKAI application. Once locally tested we will get these agents installed in production.

## 6. Progress

This project timeline indicates current progress and future plan for this research

Start of PhD Research work January 2020

### January 2020-January 2021

- Selecting the domain
- Background Study & research topic selection
- Literature Review- State of the Art
- Defining the Problem Statement
- Proposing a methodology
- Implementation of environment setup for data collection
- Synopsis Document

### February 2021-June 2021

- (Tool Development)
- Designing scenarios for data collection
- Executing scenarios for data collection
- Data Preprocessing

### July 2021-December 2021

- Formulation of Controlled Experiment
- Checking Insights
- Results Analysis

### January 2022-June 2022

- Post validation Survey
- Evaluation
- Extended Literature Review
- Compilation of Thesis
- Final Defense

## 7. References

- [1] Qureshi, N. A., Perini, A., Bruno, F., Irst, K., Ernst, N. A., & Mylopoulos, J. (2010). *Towards a Continuous Requirements Engineering Framework for Self-Adaptive Systems*. 9–16.
- [2] Raimondi, F., Skene, J., & Emmerich, W. (2008). *Efficient Online Monitoring of Web-Service SLAs*. 170–180.
- [3] Mens, T., Demeyer, S., Wermelinger, M., & Hirschfeld, R. (2005). *Challenges in Software Evolution*.
- [4] Khan, J. A., Liu, L., Wen, L., & Ali, R. (2019). *Crowd Intelligence in Requirements Engineering : Current Status and Future Directions* (Vol. 1). <https://doi.org/10.1007/978-3-030-15538-4>
- [5] Pacheco, C., & Garcia, I. (2009, June). Effectiveness of stakeholder identification methods in requirements elicitation: experimental results derived from a methodical review. In *2009 Eighth IEEE/ACIS International Conference on Computer and Information Science* (pp. 939-942). IEEE.
- [6] Hurrell, E., & Smeaton, A. F. (2012). *An Examination of User-Focused Context-Gathering Techniques in Recommendation Interfaces*.
- [7] Ghasemi, M., & Amyot, D. (2019). From event logs to goals : a systematic literature review of goal - oriented process mining. *Requirements Engineering*, (0123456789). <https://doi.org/10.1007/s00766-018-00308-3>

- [8] Fotrousi, F., & Fricker, S. A. (2016). QoE probe: A Requirement-monitoring Tool. In *REFSQ Workshops*.
- [9] Froehlich, J., Chen, M. Y., Consolvo, S., Harrison, B., Landay, J. A., Street, S. B., & Mateo, S. (2007). *MyExperience: A System for In situ Tracing and Capturing of User Feedback on Mobile Phones*. 57–70.
- [10] Oriol, M., Stade, M., Fotrousi, F., Nadal, S., Varga, J., Seyff, N., & Schmidt, O. (2018, August). FAME: supporting continuous requirements elicitation by combining user feedback and monitoring. In *2018 IEEE 26th International Requirements Engineering Conference (RE)* (pp. 217-227). IEEE.
- [11] Wüest, D., Fotrousi, F., & Fricker, S. (2019, March). Combining monitoring and autonomous feedback requests to elicit actionable knowledge of system use. In *International Working Conference on Requirements Engineering: Foundation for Software Quality* (pp. 209-225). Springer, Cham.
- [12] Glinz, M., & Roots, A. T. (2019). CrowdRE : Achievements, Opportunities and Pitfalls. *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*, 172–173
- [13] Maalej, W. (2015). *Bug Report , Feature Request , or Simply Praise ? On Automatically Classifying App Reviews*. 116–125.
- [14] Johanssen, J. O., Kleebaum, A., Bruegge, B., & Paech, B. (2019). How do Practitioners Capture and Utilize User Feedback during Continuous Software Engineering ? *2019 IEEE 27th International Requirements Engineering Conference (RE)*, 153–164.
- [15] Franch, X., Ayala, C., López, L., Martínez-fernández, S., Rodríguez, P., Gómez, C., Rytivaara, V. (2017). *Data-driven Requirements Engineering in Agile Projects : The Q-Rapids Approach*.
- [16] Bukhsh, F. A., Bukhsh, Z. A., & Daneva, M. (2019). A systematic literature review on requirement prioritization techniques and their empirical evaluation. *Computer Standards & Interfaces*, 103389.