Telling Us Your Needs with Your Eyes

Rongchen Li, Tong Li*
Beijing University of Technology, Beijing, China
*litong@bjut.edu.cn

***Abstract*—User feedback is valuable for the continued development of the system. Most of the user feedback analyzed today is explicit, such as app reviews. However, many users are reluctant to provide this explicit feedback, which costs them extra time. In this paper, we present an ongoing study, which investigates eye movement patterns with the purpose of automatically collecting users' implicit feedback. Specifically, we establish the semantic connection between eye movements and users' perceptions based on cognitive and empirical evidence. Our proposed approach can thus systematically assess the satisfaction of six non-functional requirements, quietly unveiling users' needs for the software they use. Finally, we design an evaluation plan in order for assessing the effectiveness of our approach.**

Keywords - non-functional requirements, eye movement pattern, empirical evaluation, eye tracking

I. Introduction

Efficiently and precisely collecting user feedback is important for understanding user requirements, based on which software systems evolve continuously. User feedback contains both functional requirements (FR) and non-functional requirements (NFR), with NFRs being more valuable. NFRs are widely recognized as playing a key role in software development. Unlike FRs, which often have structured methods to capture them, NFRs are difficult to summarize and extract. As a result, NFRs are often not well understood and fully considered in the software development process

Currently, the mainstream approach to assessing requirement satisfaction from users' feedback relies mainly on explicit user feedback, such as app store scores and reviews. Researchers sort and filter this predictive information to derive useful information from it [5], [7]. However, these methods rely on active user feedback, thus missing the needs of users who are reluctant or not good at giving feedback. In addition, studies in cognitive science showed that the participants' perception of their behaviour does not always agree with their underlying processes and intentions [2] especially when they have to detach from actual usage scenarios to make feedback. Some research has also begun to focus on methods for implicitly capturing user needs [11], [13]. These studies use conditional random field-based desire models to induce user NFRs. They require careful granularity adjustment by relevant domain experts before use and the results are reasoned by domain experts based on subjective judgment [11]. Their analysis is complex and tedious, requiring a high threshold for the average engineer to obtain useful information from them. Moreover, these methods are difficult to migrate between systems and especially difficult to deploy in practical applications. In addition, subsequent research to complement and extend them in their theoretical systems faces a greater challenge.

We believe that an eye-tracking based requirements assessment framework can effectively address the above issues, and to do so we address the following challenges in our research objectives: first, we need to identify the mapping relationship from non-functional requirements to users' eye-tracking behaviors, which is an unstudied area, and for this purpose we investigate and establish the semantic links between six typical NFRs and the above metrics, and propose a non-disruptive way to quantitatively assess the satisfaction of NFRs. Secondly, we need to summarize the mapping relationships from the existing studies of users' eye-tracking behaviors to the corresponding eye-movement metrics. We investigate and study the existing metrics of eye-movement data, based on which we establish a set of metrics that are closely related to the assessment of NFRs. Finally, we need to refine users' eye-movement patterns through experience and experiments, and try to assess the degree of satisfaction of non-functional needs through pattern inference. We plan to apply the model to the user testing phase of software development so that developers can quickly complete User Interface Testing and reduce the time and labor cost of this process. As an ongoing study, we also present an evaluation plan which will be executed to assess the effectiveness of our approach.

II. Related Work

There are two ways to obtain users' feedback nowadays, one is to get explicit feedback through their reviews, and the other is to get implicit feedback through their behaviour.

Reviews are the easiest way to obtain explicit feedback from users about the application, which contains much requirement-related information, such as bug reports, feature requests, user experiences, ratings, etc. User reviews are characterized by their huge number. A recent study found that mobile apps receive about 23 user reviews per day, while popular apps, such as Facebook, receive an average of 4275 user reviews per day [6]. A large amount of research is currently focused on identifying valuable information from unstructured and informal user comments. Various classification techniques have been developed to distinguish between types of reviews [5], [7], most of which focus on functional requirements. However, user reviews include both functional and non-functional requirements, and some studies focus on finding NFRs associated with quality characteristics from user reviews [12]. Analyzing user requirements for a system from explicit user feedback faces an unavoidable problem: users must actively express their requirements outside of product use, and this explicit feedback usually requires additional effort or time, leaving developers at risk of missing the perceptions of users who are reluctant or inept at providing feedback. In addition, because it is detached from the use of the product in the field, user feedback may be inauthentic or distorted.

The relative benefit of evaluating user needs implicitly is that it does not require the user to do anything, and everything captured is relatively objective and valid. Some studies have attempted implicit NFR evaluation to explore system-user interactions quantitatively through conditional random fields to uncover potential user needs or requirements [13]; or to infer human needs by monitoring environmental context and human behavioural context [11]. However, all these efforts are still exploratory and face problems such as not being easy to deploy and difficult to scale.

To conclude, we need a simple, straightforward and systematic way to evaluate user requirements implicitly.

III. Eye-tracking based Requirements Assessment Framework

The structure of the entire framework is shown in figure 1. The inputs to our framework are subtasks of the function that have been labelled by the developer with the key “area of interest”(AOI). In this section, we will first describe how we obtain eye-tracking metrics. Then, it is explained how we can use eye-tracking metrics to evaluate the level of satisfaction of non-functional requirements for subtasks. Finally, a method is proposed to extract several patterns from the metrics and use the patterns to more finely and comprehensively detect the user’s invisible needs and evaluate the degree of satisfaction of the non-functional needs of the software.

A. Obtaining Eye-tracking Metrics

By referring to the research methods of other eye-tracking projects, we can use several eye-tracking metrics to portray different aspects of user perception. High-level metrics are based on low-level metrics, which require more complex processing, including but not limited to computing, de-noising, etc. The higher the level, the better the representation of narrow semantics and clear characteristics, so higher weights will be given to higher-level metrics in our studies.

1) First-level metrics: The raw data that most affordable eye-tracking devices can collect is **Eye gaze point position on screen (X, Y)**. Generally, the collection of this data relies on computer vision technology to track the angle of rotation of the user’s head and eyes.

2) Second-level metrics: The eye tracker can distinguish between **Fixations** and **Saccades** by using an event detection algorithm for the location of the eye’s indicated landing point. We chose the discrete threshold-based method with excellent accuracy and robustness.

3) Third-level metrics: “Area of interest (AOI)” was introduced at this level as a tool to mark stimulus areas and extract indicators specifically for these areas. We use the components in the software as the AOI area in the study.

Fixation duration (FD): Time spent gazing at the key AOI. Previous studies have used this metric to characterize participant’s effort [10]. The metric’s value is related to the amount of effort the user puts into the AOI.

Fixation count (FC): The number of fixations in key AOI. It can also describe the amount of energy or effort that participant spend on the AOI. However, unlike fixation duration, prolonged gaze may produce only a single or a small number of fixation counts.

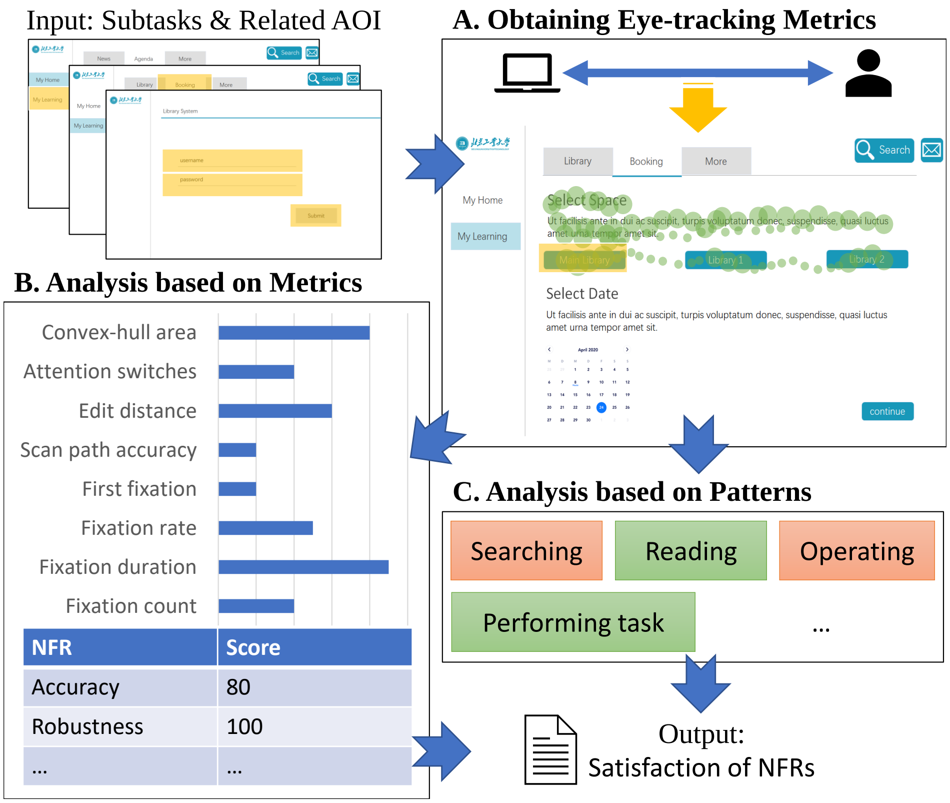
Fixation rate (FR): Ratio of the total number of fixations on key AOI to others. A lower fixation rate indicates a lower efficiency in search tasks: participant spend more effort to find relevant areas [9]. Higher rates may indicate that more effort is required to complete tasks [10].

First fixation time (FFT): Time to the first fixation in an AOI, which can indicate the attention-grabbing level of the key AOI.

4) Fourth-level metrics: In the four-level metrics, the concept of “scan path” is introduced, which is a sequence of gaze points or AOIs that describes the length and duration of fixation. Performing comparisons between scan paths allows identification and analysis of participant’s viewing strategies for solving tasks [3].

Scan path accuracy (SPA): The ratio of the number of key AOIs to the number of other AOIs among the AOIs that have been gazed at. A higher value indicates a higher level of understanding of the task by the participant [8].

Edit distance (ED): The editing cost of converting a scan path to the shortest path determined by the developer, is calculated by basic operations such as insert, replace and delete. The smaller the value,



Ut facilisis ante in dui ac suscipit, turpis voluptatum donec, suspendisse, quasi luctus amet urna tempor amet sit.

NFR	Score
Accuracy	80
Robustness	100
...	...

Figure 1. The structure of Eye-tracking based requirements assessment framework.

the more the participant's behaviour matches the developer's expectation.

Number of attention switches (AS): The total number of switches between a list of AOIs per unit time. A higher number of switches means that participant are more uncertain about the task being performed.

A convex-hull area(CA): The smallest convex set of fixations that contains all of the participant's fixation. Smaller indicates that the closer the distance between the gaze points, the less effort participant put into finding the relevant region [4].

B. Analysis based on Metrics

It is well known that non-functional requirements encompass all aspects of software design and are more difficult to define, measure, test and track than traditional functional requirements. After sifting and sorting through the non-functional requirements, we finally selected several non-functional requirements that are widely used [1], closely related to user experience and have clear characteristics. We found that once an NFR was not satisfied, some characteristic change in the user's metrics occurred. The user behavior characteristics represented by each metric (mentioned in paragraph A above) are combined with the user behavior characteristics that appear in the human-computer interaction when each NFRs is not satisfied. We used the user behavior characteristics as a bridge to correspond the eye-tracking metrics to the NFRs. Among them, some metrics respond substantially when an NFR is not satisfied, while others do not differ significantly when this NFR is satisfied or not. The correspondence between the NFRs and the changed characteristics we have identified so far is shown in figure 2. We choose two NFRs to tell how we define such a characteristic. Furthermore, how we used the eye-tracking metric to evaluate the degree of satisfaction of each NFR.

To illustrate, we detail the analysis and mapping as well as the satisfaction scores regarding the two NFRs: “Consistency” and “Ease of use”.

1) Consistency: When consistency is not met, it indicates that the system does not provide the same design pattern, meaning that elements with the same ideograms behave differently or that elements with different ideograms have the same appearance or behaviour patterns. This can be confusing for users who have difficulty spotting the element they need in the first place. The most distinctive feature of elements that cannot be found in the first place is that the first fixation at the relevant element is late, as the user is first attracted to other elements. In addition, the user will exhibit search characteristics when forced to find the correct element of interest after a failed attempt. The act of searching causes

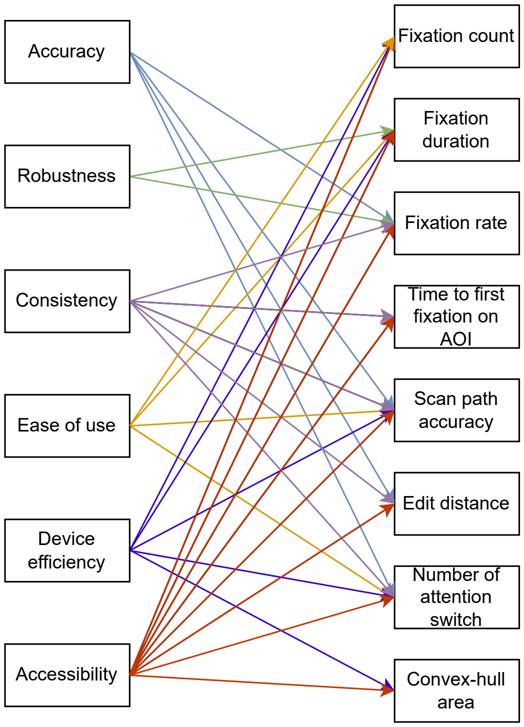


Figure 2. Association between NFRs and eye movement indicators

fixation to be scattered across other AOIs, resulting in lower fixation rate and scan path accuracy; the unpredictability of user search behaviour also leads to a significant increase in edit distance; frequent switching in AOIs increases the number of attention switching.

In summary, the metrics that change significantly when “Consistency” is satisfied and not satisfied are.

- Fixation rate
- Time to first fixation on AOI
- Scan path accuracy
- Edit distance
- Number of attention switches

However, in the search, the convex hull area is small because the user is just confused by the confusing design patterns and is clear about the general direction. So the user will search around the key AOI and the range will not be very large. In addition, once the key AOI has been found, the user faces no more difficulties with the operation. So his fixation count and fixation duration are as usual.

In summary, the indicators that do not change significantly when “Consistency” is satisfied and not satisfied are.

- Fixation count
- Fixation duration
- Convex-hull area

To summarise, fixation rate, first fixation time, scan path accuracy, edit distance, and the number of attention switches vary depending on how well the consistency is met. The worse the level of satisfaction, the lower the fixation rate, first fixation time and scan path accuracy followed, and the higher the editing distance and the number of attention switches followed. Therefore, we use the following formula to evaluate the satisfaction of this NFR. Note that we have set different weights for metrics at different levels, with the higher the level, the higher the weighting.

$$Score = \alpha * (FR + FFT) + \\ \beta * (SPA - ED - AS), \alpha < \beta$$

2) Ease of use: When ease of use is not satisfied, the service is hard for users to operate and control. Too much difficulty in operation leads to more effort and makes users sceptical and make multiple attempts. Too much effort and repeated attempts by the user will result in an unusually high amount of time and attention spent on key AOIs, so the fixation duration is long. Then, the task is so difficult to perform that the user may wonder whether the right action is the right action, thus triggering suspicion and a search for other elements around, which brings low scan path accuracy and a high number of attention switches. Because the search behavior is interspersed with the operation, the fixation count increases.

In summary, the metrics that change significantly when “Ease of use” is satisfied and not satisfied are.

- Fixation count
- Fixation duration
- Scan path accuracy
- Number of attention switches

However, because the number of fixations inside and outside the key AOIs is both large, the fixation rate is not significantly higher, and the search area is around key AOIs. Hence, the convex-hull area is small. In addition, all search behaviour comes after the user has found the relevant elements which difficult to manipulate. The user is correctly directed to the relevant element to start the action, so the first fixation is early as normal, and the editing distance is short.

In summary, the indicators that do not change significantly when “Ease of use” is satisfied and not satisfied are.

- Fixation rate
- Time to first fixation on AOI
- Edit distance
- Convex-hull area

To summarise, only the fixation count, fixation duration, scan path accuracy and the number of attention switches have changed. For the fixation duration and number of attention switch parameters, lower values indicate better satisfaction of this NFR; for the scan path accuracy, higher values indicate better satisfaction of the NFR. We use the following formula to evaluate the degree of satisfaction with ease of use. The coefficient assigned to fixation duration is relatively small because it is a lower level metric. In contrast, the coefficients assigned to scan path accuracy and attention switch count are higher because they are higher-level metrics.

$$\begin{aligned} Score = \alpha * (-FC - FD) + \\ \beta * (SPA - AS), \alpha < \beta \end{aligned}$$

C. Analysis based on Eye Movement Patterns

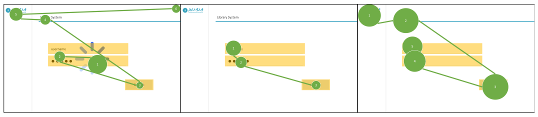
The analysis above demonstrates that many fixed patterns recur across multiple NFRs. We carry out further work to abstract these patterns from the eye-movement metrics to help us better evaluate non-functional requirements. We use the idea of a variable-length sliding window to analyze the multiple metrics mentioned above at a finer granularity on the timeline of the user's task completion. From the point of view of user perception, the user's behaviour is always made up of multiple patterns when each NFR is not satisfied. We take two patterns as an example, one for "searching" and one for "performing task", which we will describe in more detail below. These two models have set the stage for more models to be proposed in the same way to provide a more granular analysis of the NFR assessment.

1) Searching: Through the analysis experience, we used low fixation rate, low scan path accuracy and a high number of attention switches as a pattern. When they appear in combination in a window of time, they represent a search behaviour used by the user, as the fixation landing point is bound to wander between multiple AOIs when searching. There are multiple NFRs that users exhibit search behaviour during task execution when they are not satisfied. For example, when the ease of use is not satisfied, the user intersperses the search behaviour in the middle of the task execution to exclude the possibility of the task being difficult due to his actions; when consistency is not satisfied, users exhibit search behaviour at the beginning to find the key AOI; when device efficiency is not satisfied, the user starts searching for changes on the whole page due to boredom of waiting, etc.

The searching pattern appears in different positions in the different cases that each NFR is not satisfied, which means different for each NFR. For example, in the case of consistency, the search pattern is the core behavioural feature. Using the captured data, we can further analyze which elements conflict with the key AOI design pattern from the searched object, which confuses the user. However, in the case of device efficiency, the search is just a by-product when the NFR is not satisfied, so we do not need to care about the specific object.

2) Performing: When the fixation rate is extremely high, the fixation duration is extremely long, the scan path accuracy is extremely high, and the convex-hull area is extremely small in a period of time window, we can conclude that the user is completing the sub task's goal. Regardless of the existence of NFR unsatisfaction, the execution task pattern will always occur because, for each subtask, completion is a mandatory path to the next subtask. When either consistency or ease of use is not satisfied, the user can only have two behavioural patterns, searching and performing. However, the difference is in the order of the patterns: the user behaviour when consistency is not satisfied is "Searching" - "Performing", while the ease of use is not satisfied by the user behaviour is "Performing" - "Searching" - "Performing".

3) More Patterns: Assessing the degree of NFR satisfaction with pattern contains the richest amount of information and the highest degree of flexibility. We can also propose more patterns by the same research method. For example, by adding the constraints of small convex hull areas and long gaze times to the search pattern, it becomes a more specific "learning pattern". It will be shown by the user when accessibility is not satisfied, meaning that the user has no knowledge of the elements on the interface and needs to learn and familiarize himself with the whole interface on a large scale to have enough information to help him take the next step. In the future, we intend to give the highest weight to patterns as the highest level, the clearest representation and the most credible metric among all metrics. When there are enough patterns and detailed enough, we can describe the satisfaction of each NFR entirely using patterns to accurately and finely portray the satisfaction of NFRs.



(a) Searching. (b) Performing. (c) Learning.

Figure 3. The raw eye-tracking data of several patterns, the yellow fluorescent area represents the critical AOI area, the dot represents fixation, the longer the fixation duration, the larger the dot.

IV. Prototype

We have now completed the data capture part and the software model to be tested in the whole system. The hardware device uses a Tobii Eye Tracker 5 gaming eye-tracker, based on the Stream Engine API it provides to obtain raw x and y data from the user's gaze point. The software under test communicates with the data capture platform via a back-end application that records the raw eye-tracking data acquired by the hardware device and the execution of subtasks on the front-end and the real-time position of the components, thus providing the basis for later analysis. Building on the above, we are working on implementing a data analysis system, including but not limited to extracting high-level data from raw data, obtaining NFR satisfaction of subtasks from metric features, etc.

V. Evaluation Plan and Preliminary Results

Currently, we can still only perform a rough qualitative analysis of all metric models and can only speculate how high or low each metric is. To obtain exhaustive, systematic and quantitative models, we need to design experiments to collect eye movement data corresponding to each NFR during real user tests.

Selection of Participants. The participants in the control experiment will be recruited from among people who had some basic knowledge of computer use. Pragmatically, to ensure that the number of participants led to relatively reasonable conclusions, we recruited 20 students majored in computer science and technology to participate in the experiment.

Experimental Materials and Tasks. We designed a simulated university library system based on a web application. The software itself has functions such as information query and seat reservation . Developed the ability to insert intentionally bad designs on it in real-time, we use the application as the target software for evaluating the six typical NFRs described above, then let participants use the software and focus on evaluating the library seat reservation function. We will divide the entire function usage flow into a number of seven subtasks, such as “Login to the system” and “Navigate to the function page”. As participants complete each subtask, we will use the eye-movement data collection system mentioned above to collect their eye-movement data.

Experimental Design. We divided the participants into two groups, the experimental and control groups, with ten people in each group. Based on the self-developed test software for inserting bad designs described above, we designed test tasks for different NFRs in different subtasks covering the six typical NFRs we mentioned. In the test for the experimental group, we turned on intentional bad designs for the participants to trigger their NFRs. During the experiment, we asked participants to use the “thinking aloud” method to verify that the intentionally poor design does induce their corresponding NFR. These designs are intentionally staggered so as not to interfere with each other and give the participants time to recover from the previous bad designs caused by confusion and dissatisfaction. In the control group, we did not turn on any intentionally bad designs, and the data obtained from them will be used as the standard data for completing each subtask.

Validation and Improvement. We will compare the differences between the collected experimental data and the control data, analyzing whether the differences between them are consistent with the characteristics of our proposed indicator model. Finally, in conjunction with these data, we hope to improve our model by upgrading the judgement of each metric characteristic from a high or low qualitative to a specific scale. This will allow our model to be more precise in its judgements.

Results Analysis. According to the experimental plan above, we made a toy experiment. Some of the results of the experiment are shown in figure 4. For “consistency”, the experimental results show the correspondence analyzed in section 3b above. Fixation rate, first fixation time and scan path accuracy were negatively correlated with the degree of satisfaction of ”consistency”. Editing distance and the number of attention switches were negatively correlated with the satisfaction of “consistency”. In contrast, the data of the other indicators in the experimental group do not do at least the median and the mean at the same time greater or smaller than the data of the control group, that is, there is no significant difference between the two groups, so we will ignore them in the correspondence.

VI. Discussion

Interaction between NFRs. It is well known that NFRs may be related to each other. This also means that in real-world applications, they may affect each other to the extent that they do not receive separate feedback from the user. This is an objective problem, and many studies focus on the interactions between NFRs. However, it is not within the scope of our research. In this work, we try to construct the most prominent and simple requirement scenarios that independently link metrics and NFRs according to the definition of NFRs. In our subsequent work, we will try to consider the relationship between NFRs and refine the mapping by combining research results on related topics.

Extension of use cases. This project uses a hardware device that is essentially just a camera sensor with an IR fill light to obtain all the required parameters. This means that the framework can be deployed on a large number of devices (PCs with Windows Hello face authentication, Apple mobile devices with Face ID and Android mobile devices with face authentication). With full adherence to the principle of openness and voluntariness, and with proper handling of privacy regulations, we can look forward to an approach that will get feedback from numerous remote, unknown users implicitly. This will provide a new possibility for feedback on user requirements for software systems.

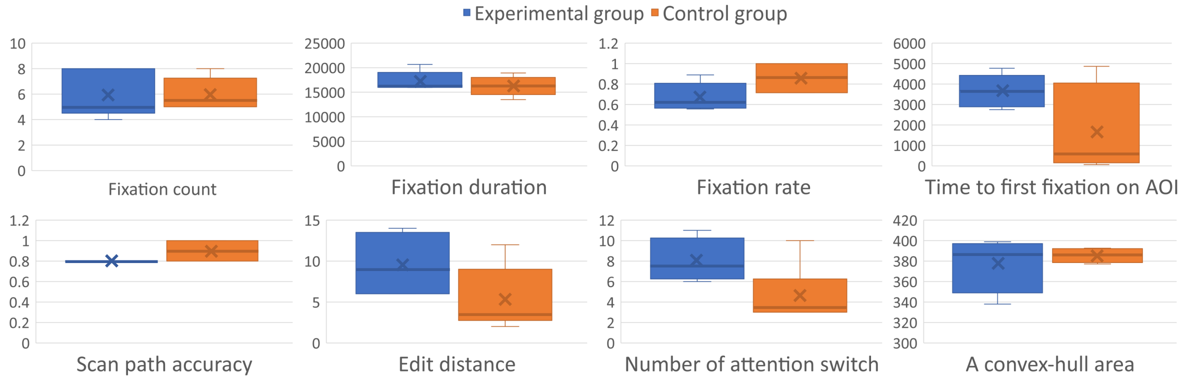


Figure 4. The experiment result of "Consistency".

VII. Conclusions and Future Work

At a time when non-functional requirements are gaining importance, and there are more ways to extract and mine them, we still feel that we can extract increasingly valuable information for analysis from the actual users of the product. In this paper, we present eye-movement metrics that can effectively distinguish non-functional requirements and use several examples of non-functional requirements to demonstrate how the system works, focusing on providing a way of direct observation at the user that can be used to describe and inform the study of non-functional requirements.

Our long-term vision is to build a systematical, highly compatible, and customizable evaluation system that, together with appropriate development techniques, facilitates small and medium-sized developers and provides new inspiration for larger ones. This paper is the starting point for an integrated and structured approach. In future work, we will give methods to identify quantitative analysis for each dimension of eye-tracking metrics to be compatible with more non-functional requirement classifications. We also intend to validate the proposed techniques in the open-source community, combining real projects with detection systems and experimentation to identify potential oversight issues and improvements to our approach.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (No.62162051, 61902010).

References

[1] Dictionary of Non-functional Requirements of Business Process and Web Services.

[2] Dave Binkley, Marcia H. Davis, Dawn Lawrie, Jonathan I. Maletic, Christopher H. Morrell, and Bonita Sharif. The impact of identifier style on effort and comprehension. *Empirical Software Engineering*, 18:219–276, 2013.

[3] Teresa Busjahn, Roman Bednarik, Andrew Begel, Martha E. Crosby, James H. Paterson, Carsten Schulte, Bonita Sharif, and Sascha Tamm. Eye movements in code reading: relaxing the linear order. In *International Conference on Program Comprehension*, 2015.

[4] Joseph H. Goldberg and Xerxes P. Kotval. Computer interface evaluation using eye movements: methods and constructs. *International Journal of Industrial Ergonomics*, 24:631–645, 1999.

[5] Walid Maalej, Zijad Kurtanović, Hadeer Nabil, and Christoph Stanik. On the automatic classification of app reviews. *Requirements Engineering*, 21(3):311–331, September 2016.

[6] Dennis Pagano and Walid Maalej. User feedback in the appstore: An empirical study. In *2013 21st IEEE International Requirements Engineering Conference (RE)*, pages 125–134, July 2013. ISSN: 2332-6441.

[7] Sebastiano Panichella, Andrea Di Sorbo, Emitza Guzman, Corrado A. Visaggio, Gerardo Canfora, and Harald C. Gall. How can i improve my app? Classifying user reviews for software maintenance and evolution. In *2015 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 281–290, September 2015.

[8] Razvan Petrusel and Jan Mendling. Eye-tracking the factors of process model comprehension tasks. In *Conference on Advanced Information Systems Engineering*, 2013.

[9] Alex Poole, Linden J. Ball, and Peter Phillips. In Search of Salience: A Response-time and Eye-movement Analysis of Bookmark Recognition. In Sally Fincher, Panos Markopoulos, David Moore, and Roy Ruddle, editors, *People and Computers XVIII — Design for Life*, pages 363–378, London, 2005. Springer.

[10] Zohreh Sharafi, Bonita Sharif, Yann-Gaël Guéhéneuc, Andrew Begel, Roman Bednarik, and Martha E. Crosby. A practical guide on conducting eye tracking studies in software engineering. *Empirical Software Engineering*, 25:3128–3174, 2020.

[11] Peng Sun, Jingwei Yang, Hua Ming, and Carl K. Chang. A Multi-layered Desires Based Framework to Detect Users' Evolving Non-functional Requirements. In *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, volume 01, pages 28–37, July 2018. ISSN: 0730-3157.

[12] Tianlu Wang, Peng Liang, and Mengmeng Lu. What Aspects Do Non-Functional Requirements in App User Reviews Describe? An Exploratory and Comparative Study. In *2018 25th Asia-Pacific Software Engineering Conference (APSEC)*, pages 494–503, 2018. ISSN: 2640-0715.

[13] Haihua Xie, Jingwei Yang, Carl K. Chang, and Lin Liu. A statistical analysis approach to predict user's changing requirements for software service evolution. *Journal of Systems and Software*, 132:147–164, October 2017.