Enhancing User Experience using Machine Learning

Sumaiya PK

SAP Labs, Whitefield, Bangalore, Karnataka, India.
Sumaiya.p.k@sap.com

Abstract **– In business solutions, what differentiates the best product from the rest is "Customer Satisfaction". Most businesses understand the importance of customer satisfaction and invest efforts in it. However, the implementations in place today are time-consuming and often don't solve the core problem. This paper aims to presents a solution that addresses the gap observed in the software implementations of today. The main idea is the lossless capture of user feedback to identify the negative points and solve them at a swift pace. The proposed solution will use 'Clickstream' to capture all the actions performed by the user and feed that data into a Machine Learning Algorithm to identify areas of improvement. User's actions will be captured as events using 'Clickstream' along with the time taken to perform the action, the location and demographic details of the user. This data will be captured only upon the user's consent. Such an implementation also accounts for the diversity in the usage of the solution and identify problems backed by data.**

Keywords - Clickstream, Machine Learning, MongoDB, User Activity Tracking.

I. INTRODUCTION

A Software solution serves its true purpose when it answers the problem it sets out to solve. There are innumerable solutions available in the market today that tackle the same problem. What differentiates the best solution from the rest is the experience a user has while using the solution. How an end user feels about a solution should be a key factor of consideration since it's the end user who interacts with the solution. Today, however, the feedback is collected through intermediaries. The customers share the feedback with the consultants, consultants share their interpretation to the lead, who then forwards it to the product owner, who finally shares it with the developers. In such a process, the feedback provided by the end-user is often lost. A user who finds using a solution difficult has lower productivity at work - a task that should be completed in 5 hours may end up taking 7 hours or more. Herein lies a unique opportunity. Imagine a scenario where the system detects that a user is struggling with a solution and collects relevant data and notifies the developer concerned with it. The development team can then look at the data and solve the problem. The end-user now has a richer experience of using the UI without having to invest additional effort. This improves the user engagement which drives up the productivity thereby improving the business case. From a technology standpoint, we have the tools in place to build the first rendition of this experience. Upon the user's consent, we track the user activity and feed it to Machine Learning models that will identify pain-points observed by the users. Based on the insights, the model can alert the developers through a report. The product team can then work on improving the user experience by keeping the model generated report as a reference.

II. LITERATURE REVIEW

To establish the relevance of the solution proposed, we must first identify the shortcomings observed in the solutions present today. This section lists the usability issues that are observed often.

A) Lack of appropriate name for UI Element

A user understands the role of a UI element by referring to the text that adjoins the element. Many times the name given to a UI element is not self-explanatory. From a development point of view, there is a tendency to make the UI fancy and use fancy vocabulary. However, we deal with a variety of customers spread over different regions consisting of users of different age group and different dexterity. Thus, analysis of what naming format is appropriate is quite difficult. Machine learning can help us analyze and decide on a better naming convention [8].

B) Self-Understanding

The UI elements should be clearly visible, spacious and self-understandable. The icons used and the color, fonts, etc., should be connected to the real world for easy grasp [10]. One can observe that in cases involving complex logical or business processes the UI becomes increasingly and sometimes excessively complicated. This leads to the users investing a lot of time in understanding the flow and finding the required UI elements on the Screen.

C) Varying perceptions of developer and customer

It is an extremely well-known fact that the expertise of a beginner is not the same as an experienced person. We are familiar with the things that we use often. Similarly, when any developer works on an app, they become familiar with the terms they use in a UI. Relying on such knowledge introduces a bias into the application. Customers use different apps at different times. In word document while performing the operations like copy and paste, generally experienced person use Ctrl C and Ctrl V, that doesn't imply that everone knows this shortcut and this should not be made the only option available to the end user. We need to provide traditional methods as well. Hence, we need to know the customer's level of understanding to develop user-friendly apps.

D) Quick Turnaround time

We all understand the quote 'Time is money'. A user expects issues to be resolved on time. However, the working model in place today has a lot of checkpoints in place. With multiple stakeholders involved such as Project Manager, User Representative(s), Developer(s), Support team, the time available for response is depleted [9]. A significant amount of time is spent by the brokers to identify the issue through emails and calls – an activity that can easily be avoided.

III. HARDWARE AND SOFTWARE USED

In this section we will be focusing on understanding the key terminologies used in the solution.

A) Clickstream

The process of recording and analyzing the user actions such as mouse movements, scrolling, refreshing etc., in any Web browser or any software application is called Clickstream. The user interaction data is stored in a log history, allowing us to see the order and details of user activity When the user clicks/types anywhere in the application, the user's activity will be recorded. The activity log will also collect the details about the host device, the browser used, time taken for loading, the page/view in the browser/application, number of pages viewed, frequency of the pages viewed, different features used, User details, etc.. This data will provide hints on a user's comfort in using the application and identify potential pain-points where the user finds using the software difficult.

B) Machine Learning

Machine Learning(ML) involves the process of training the machine. In this process, we try to convert experience into Knowledge. The Machine is trained using large amounts of data, this data acts as an input. The input will be added to a training algorithm, which analyzes the data and recognizes patterns. The outcome will be the knowledge [2].

C) Why machine learning

A software solution is used by many users providing huge amount of data. Each user interacts with the software differently. For a user, the experience of working is influenced by a wide range of behavioral parameters such as emotional, experiential, affective and aesthetics [11]. One could be tempted to devise a model that analyzes the behaviour of each user independently and suggests improvements specific to the user. Such a model, however, is not efficient and in the context of thousands of users becomes unmanageable. Additionally, we need to be mindful of the factors at play here. Users of a software are generally spread all over the world, working in different time zones, having different levels of experience, different knowledge level, etc. [1]. Even within the same continent users belonging to different country exhibit different behavior. This applies to different states in the same country and different cities in the same state. All these combinations are based on just one parameter - geographic locations. We can include many other parameters such as experience level, requirement, type of data base needed, etc., but on doing so we soon realize that this isn't a sustainable model. Hence, we need a framework that is capable of adjusting to the ever-changing customer scenario. Machine learning algorithms can learn continuously and provide analysis.

D) MongoDB

MongoDB is a document oriented database. The data will be stored in JSON style documents. Different documents come together to form a Collection. Each document is independent of the other in terms of fields, content, size etc. MongoDB is easy to scale which makes it extremely apt for use-cases involving Big-Data[6].

E) User Activity Tracking

With the help of Clickstream, we can record the activities performed by the user. This tracking will be done with the help of Clickstream. The data captured will include the URL, Application Name, the pages viewed, time spent on a page, texts entered/edited/deleted in Text boxes, checkbox etc..

The data will be captured adhering to the Data Privacy Rules as agreed on the Non-Disclosure Agreement (NDA).

F) Sample Data Calculation

Note: Calculating the data based on certain assumptions.

Assuming 10000 users of one application

One application requires around 200 clicks (User activity)

So, we have 10000*200 = 2000000 data points in one day

One month 2000000 * 30 = 60000000 data points in one month:

Just in ERP there are around 12 major fields; considering each field as one app:

We have 60000000 * 12 = 720000000 data points in one month:

Then we have 720000000 * 12 = 8640000000 data points in one year.

IV. PROPOSED SOLUTION

The process of user feedback involves many mediators.

Customer → Customer Lead → Customer manager → Consultant →Consultant lead→Consultant manager →Product Owner →Developer Team Lead → Developer

In this process of feedback everyone explains their understanding to the next level. In the process, the actual feedback might get corrupted or modified.

Hence the suggestion here is to make user experience better based on data collected at the source. While the customer is using the app, all the activities are recorded in background and added to a Machine Learning algorithm. This algorithm will help developers identify any unusual behavior in the data. The machine will learn using the data fed, and learn to identify unusual behavior. E.g.: consider one user using an application where there is a list of tasks, and user needs certain amount of time to complete the task. Now when there is an issue, the time taken will be much different compared to the normal pattern. Machine Learning algorithm will help us identify this outlier and show the cause of it.

Note: Any observation that appears to be inconsistent when compared to the remaining data set available is termed as an Outlier.

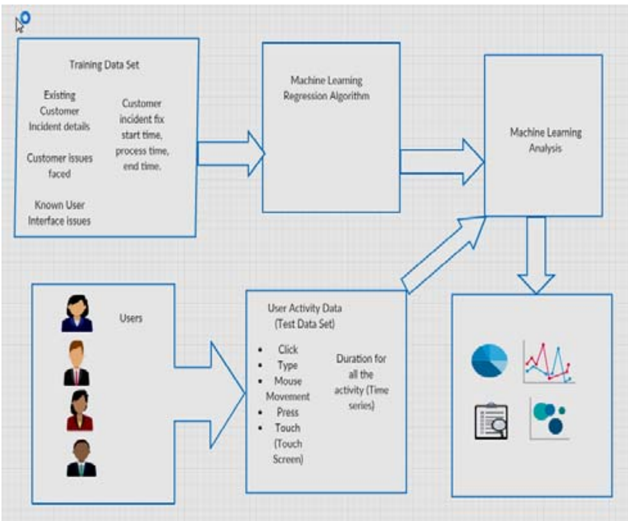

Fig. 1: Basic Architecture Diagram

TABLE I. LIST OF USER DATA THAT CAN BE RECORDED

User ID	App Name	Page/View Name	Control ID	Event	Time
001	Cloud Reporting Demo	_page0	_button0	press	2017-09-13 :14:07.19
001	Cloud Reporting Demo	_page0	_xmlview0--userid_input_id	Live Change	2017-09-13 :14:08.21
001	Cloud Reporting Demo	_page0	_xmlview0--userid_input_id	Change	2017-09-13 :14:09.42
001	Cloud Reporting Demo	_page0	_button1	press	2017-09-13 :14:19.37
001	Cloud Reporting Demo	_page0	_xmlview1--userid_input_id	Change	2017-09-13 :14:20.33

In the process of tracking the User Activity, we need to analyze what data is needed to fulfill our outcome. Table 1 shows the sample data that can be collected and used as training data set into Machine Learning Algorithm.

Of the various algorithms available, the scenario determines which one is to be used. In the process of analyzing User behavior, we use some concepts of Distance based Algorithm of detecting outliers [18].

There are 2 major datasets in ML Process.

A. Training Dataset
B. Test Dataset

A) Training Dataset

In this step, we input all the existing experience data which is the training data for the Machine. Making use of all the existing experience the Machine will be trained to analyze different patterns and perform set of grouping.

In this scenario we use existing tickets(issues raised by end user), user activity details etc to train the algorithm. With the help of this data grouping will be performed by using time as one of the parameters. By analyzing all the data, major grouping can be done which depicts the approximate time consumed to perform different sets of activities.

Consider User Activities as

$$A_1 \;, A_2, A_3, A_4 ....A_n$$

In Fig. 2 the User activity data with respect to time consumption is shown. The Graph shows different users performing activities consuming certain amount of time.

The activities which consume approximately similar time to complete will be grouped into one.

Hence, Group1(G_1) will have set of activities that fall into certain time range.

$$G_1 = \{A_1, A_2, A_5, A_8, A_{67}, A_{102},.......\}$$

Similarly, we will have $G_1, G_2, G_3,.....G_n$.

In Fig. 3 the grouping of different activities is shown by using color coding. Each color depicts different group.

e.g. The Red color depicts the activities which consume much more time when compared to other colors.

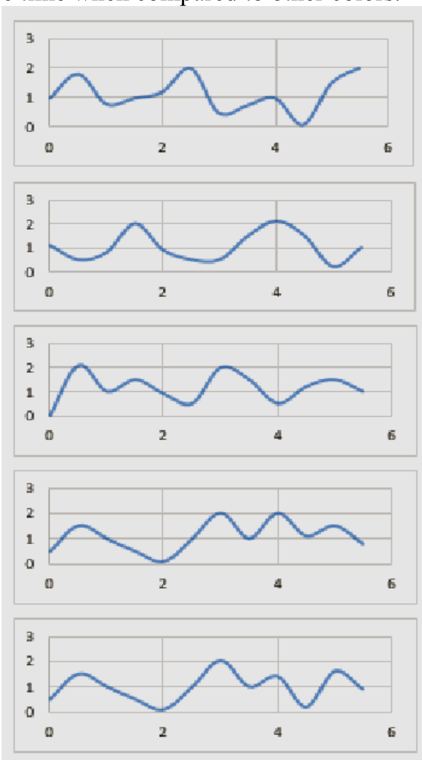

Fig. 2. Training dataset of User Activity(event) against Time

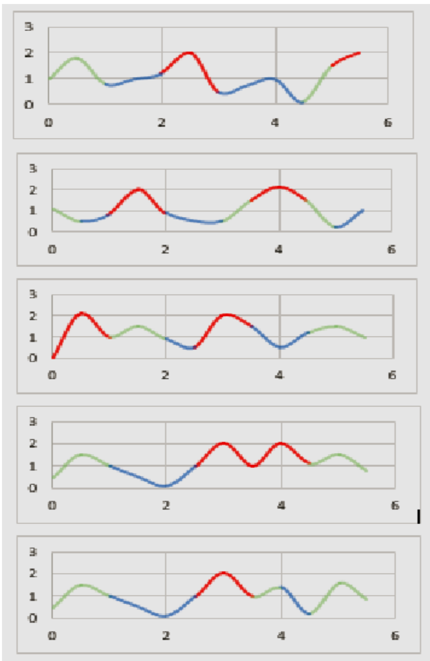

Fig. 3. Grouping of Training dataset of User Activity(event) against Time

B) Test Dataset

Once the analysis of training dataset is accomplished, we input the Actual data to obtain results.

Consider User Activities represented as A_1 to A_n.

Time Consumed for each Activity can be represented as T_1 to T_n

Initial mapping of activity to the respective group will be performed.

Considering Activity A_1 to A_{10} belongs to G_1.

Time consumed for each Activity will be as

$$G_n TA_n = G_n (T_{n+1} - T_n)$$

e.g. $G_1 TA_1 = G_1 (T_2 - T_1)$ - Time consumed for completing Activity1 which belongs to Group1.

$G_1 TA_2 = G_1 (T_3 - T_2)$ - Time consumed for completing Activity2 which belongs to Group1.

Approximate Time for each group is provided by the Analysis which can be represented as $G_1 T_{Approx}$

The difference of $G_1 T_{Approx}$ with data $G_1 TA_1$ to $G_n TA_n$ will be calculated which can be represented as Check C.

$$G_n C_n = G_n T_{Approx} \sim G_n TA_n$$

e.g. $G_1 C_1 = G_1 T_{Approx} \sim G_1 TA_1$

$$G_1 C_2 = G_1 T_{Approx} \sim G_1 TA_2$$

Then the Cluster is made of all the data for $G_1 C_1$ to $G_n C_n$. Any data which has more than allotted time difference will be considered as an Outlier.

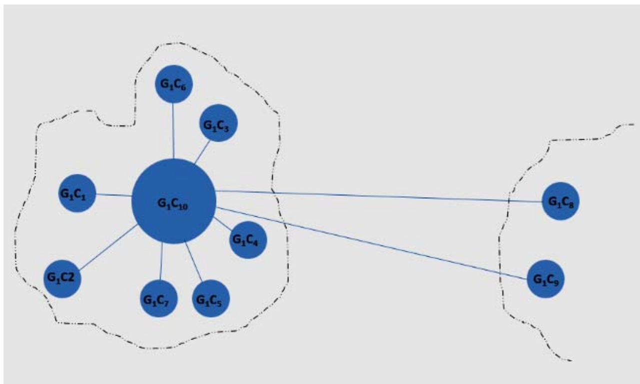

Fig. 4. Outlier detection of unusual user behavior

DOI 10.5013/IJSSST.a.20.05.01 ISSN: 1473-804x online, 1473-8031 print

V. IMPLEMENTATION PROCESS

Key steps involved

- Implement custom event listener
- Store event data in a repository
- Feed the event data to our ML Algorithm
- Analysis of output produced by ML

These steps are described in detail below.

A) Implement custom event listener

This step involves creating a custom event listener for the purpose of capturing all the data related to a User activity. Table 1 provides a probable list of interesting data points, which can be extended to include more data points based on requirement.

B) Store event data in a repository

Event data captured by the event listener should be stored in a repository. Accumulation of this data would lead to more accurate results provided by the ML Algorithm. Repository should be interchangeable to allow for quick changes in database technology e.g. Switching repository from standard SQL to MongoDB.

C) Feed event data to our ML Algorithm

Machine Learning Algorithm will process the data stored in the repository. We will make use of Regression analysis of ML and perform outlier detection, which will help us identify the event where the user is facing an issue

D) Analysis of output produced by ML

The outcome obtained by the ML can be used as needed. The time series captured along with the event will point to the exact issue faced by the user (described in detail in Scenario 1). And the analysis of number of users with respect to the event/controller will help us identify the most used features in an application which will be used for future enhancements (described in detail in Scenario 2).

VI. RESULTS AND DISCUSSIONS

The test data from the Customers(Users) are uploaded in the ML algorithm and the outcome is described in detail in the below two scenarios.

A) Scenario 1: Machine Learning Analysis

Machine Learning has the ability to find different patterns which provide valuable analysis by using large data sets. Our scenario requires the need to identify an issue. Machine is trained with the previous experience (Dataset), so any data other than the usual pattern will be identified. We make use of Novelty Detection where the machine will determine unusual pattern given a set of past experience data. The term unusual is very subjective, hence the approach here can be that each observation will be given some rating based on a degree of novelty [3]. Pattern recognition is one of the application of Novelty Detection. Pattern recognition focuses on recognizing different patterns from a given set of data. Using these concepts the analysis based on assumed data is described. The Graph in Fig 5 shows the time taken by a user to perform different activities. User1 takes 1 minute to complete activity 1, 2 minutes to complete activity 2 and so on. Which means on an average User1 takes 1.5 minutes per activity. But when it comes to activity 8 User1 takes 13 minutes to complete just one activity. This is clearly visible in the chart shown in Fig 5. Upon plotting the graph for other users, we notice this higher than average time consumption for other users as well. Fig 6 shows the user behavior with respect to different controls used in the UI. With such a graph, it is easy to analyze other parameters as well which is shown on hovering. This consolidated data will be sent directky to the development team. Now, the developer can easily figure out the area of trouble as observed by the customer and work towards providing a fix in an upcoming release. This proactivity helps in optimizing the human performance and improving the user satisfaction [12]. This also leads to the liklehood of solving an issue before the customer lodges a formal complaint through a ticket, saving time and money for everybody involved.

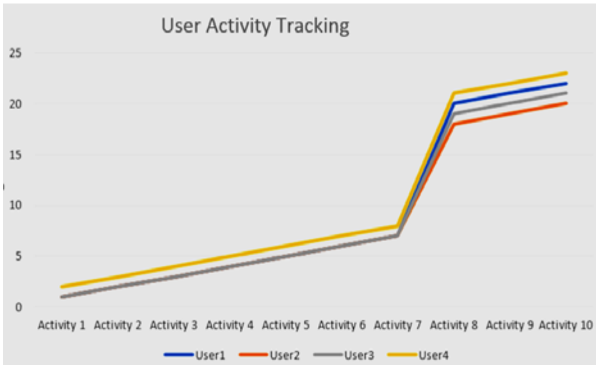

Fig. 5. Plot of User Activity(event) against Time

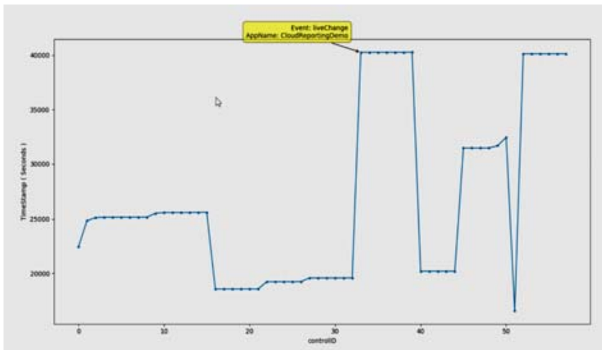

Fig. 6. Plot of User Activity (Control ID) against Time

B) Scenario 2: App usage Analytics

Customers use many applications and features across the world. We can use the past behavior of the customer to obtain a pattern of the most used features. This analysis will help predict the features that can enhance the end-user experience. The patterns are obtained based on the behavior of similar users; hence it is collaborative in nature. This analysis is termed as collaborative filtering [4]. With this honest feedback development team, will also know the need and choice of the customers. By plotting the usage of different apps by the customers, we can identify which app is used by most customers, allowing the development team to plan enhancements on the same. In Fig. 7, it's very clear that App1 has more number of users. So, the development team can plan more enhancements for the same. Similarly, App3 and App4 have a smaller share of users, which indicates that there might be an issue.

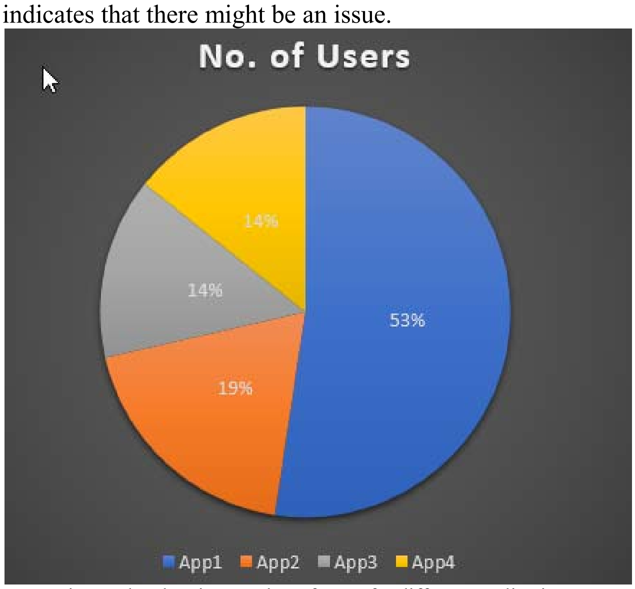

Fig. 7: Plot showing number of users for different applications.

VII. CONCLUSION AND FUTURE WORK

- Performance – The Performance of the Customer system will get affected when the tracking data hits the backend very often. Hence, intervals of capturing data should be decided based on the Performance needed.
- Scaling – In the Process of tracking the user activity, the controllers developed should be scalable so that any number of features can be tracked.

More Scaling → better performance for ML analysis
More Scaling → Reduced System Performance

The analysis obtained from the ML will help us solve various common usability issues. In the first iteration of the proposed solution, the development team will be required to infer actions from the results published by the ML model - developers will be required to analyze the ML outcome and decide which issues need to be fixed. In future, this process can be automated. The ML analysis will exhibit different patterns that highlight unusual behavior of the user while using any Website/Web Application/On-Premise products etc., We can integrate an automated voice recorded system or a pop-up help message in the User's Screen, which will be triggered when any such pattern is recorded. With this the user, can type or speak out the issue faced and with help of AI we can show a set of possible solutions and important links to refer. Doing so would lead to a situation in which the customer incidents would be raised only when the issue faced by the user is not resolved by the AI system.

ABBREVIATION

AI-Artificial Intelligence
DB- Data Base
JSON- Java Script Object Notation
ML- Machine Learning
NDA-Non-Disclosure Agreement
SQL - Sequential Query Language
UI- User Interface

REFERENCES

[1] Shai Shalev-Shwartz and Shai Ben-David "When Do We Need Machine Learning" in *Understanding Machine Learning from Theory to Algorithms*, New York: Cambridge University Press, 2014, pp. 20-24

[2] Shai Shalev-Shwartz and Shai Ben-David "Introduction" in *Understanding Machine Learning from Theory to Algorithms*, New York: Cambridge University Press, 2014, p.19

[3] Alex Smola and S.V.N. Vishwanathan "A Taste of Machine Learning" in *Introduction to Machine Learning*, Cambridge: Cambridge University Press, 2008, pp.7-12

[4] Alex Smola and S.V.N. Vishwanathan "A Taste of Machine Learning" in *Introduction to Machine Learning*, Cambridge: Cambridge University Press, 2008, p.4

[5] SAP, SAPUI5:UI Development Toolkit for HTML5, *Extending Applications*. [Online]. Available: https://help.sap.com/saphelp_uiaddon10/helpdata/en/40/7feaf830c94e4c9de48ce08adabd1c/frameset.htm

[6] Tutorials Point, MongoDB. [Online]. Available: https://www.tutorialspoint.com/mongodb/index.htm

[7] Ben Shneiderman and Catherine Plaisant "Design Issues" in *Designing the User Interface,* New York: Pearson Edition,2005

[8] Ben Shneiderman, Designing the User Interface, 3rd ed., Addison-Wesley, 1990.

[9] Context of Use Analysis, *Usability Body of Knowledge*. [Online]. Available: http://www.usabilitybok.org/cognitive-walkthrough

[10] Bhaskar N U, Prathap Naidu P, Ravi Chandra Babu S R and Govindarajulu P. "General Principles of User Interface Design and Websites," *International Journal of Software Engineering (IJSE), vol.2, Issue.3. 2011*

[11] Law, E. L. -C., Roto, V., Hassenzhal, M., Vermereen, A. P. O. S. and Kort, J. (2009). Understanding, scoping and defining user experience. *Proceedings of the 27th International Conference on Human Factors in Computing Systems – CHI 09*, 719. http://doi.org/10.1145/1518701.1518813

[12] Nigel Bevan, "What is the difference between the purpose of Usability and User Experience Evaluation Methods?"

[13] Yao L. et al. (2014) Using Physiological Measures to Evaluate User Experience of Mobile Applications. In: Harris D. (eds) Engineering

Psychology and Cognitive Ergonomics. EPCE 2014. Lecture Notes in Computer Science, vol. 8532. Springer, Cham.

[14] Graziotin, Daniel & Wang, Xiaofeng & Abrahamsson, Pekka. (2015). The Affect of Software Developers: Common Misconceptions and Measurements. 10.1109/CHASE.2015.23.

[15] E. Fox, C. Guestrin, "Machine Learning: Regression, University of Washington". [Online] Coursera, Available: https://www.coursera.org/learn/ml-regression

[16] Y. S. Kim, B. J. Yum, J. Song and S. M. Kim, "Development of a recommender system based on navigational and behavioral patterns of customers in ecommerce sites.", in *Expert systems with Applications*, 2005, pp-381-393.

[17] J. Lee, M. Podlaeck, E. Schonberg and R. Hoch, "Visualization and analysis of click stream data of online stores for understanding web merchandizing.", in *Data Mining and Knowledge Discovery*, 2001, pp.58-80

[18] Edwin M.Knorr and Raymond T. Ng. Algorithms for mining distance-based outliers in large datasets. *In Proc.24th VLDB*, pp.390-400 1998.