Olga Danilova

Optimizing User Experience through Analysis of the User Activity Patterns

Master's Thesis
in Information Technology
July 6, 2021

University of Jyväskylä

Faculty of Information Technology

Kokkola University Consortium Chydenius

Author: Olga Danilova
Contact information: o-danilova@hotmail.co.uk
Phonenumber: 040-0943 631
Title: Optimizing User Experience through Analysis of the User Activity Patterns
Työn nimi: Optimizing User Experience through Analysis of the User Activity Patterns
Project: Master's Thesis in Information Technology
Page count: 56+2
Abstract: The User Experience has gained a lot of interest in the Web development world lately. The need to hear the user during the process of application design and development has become obvious. It is a general practice of the leading software solutions nowadays to apply agile solutions in the development process and center the design around the user. The possibility to achieve user centred design without direct interaction with the user group is a challenge, which can be met by using the user activity patterns for learning the User Experience with the application indirectly. Server logs can be a sufficient source of data for such research. An opportunity to learn User Experience from the User Activity patterns, emphasise the need to hear the user and learn from the user, a chance to present a solution for reducing the gap between the developer and the end user are the key goals of the given study.
Suomenkielinen tiivistelmä: -
Keywords: Data Processing User Experience User Activity Patterns Agile Software Development User Centered Design Sequential Patterns Analysis
Avainsanat: -

Preface

"... if people are made safer, more comfortable, more eager to purchase, more efficient - or just plain happier - by contact with the product, then the designer has succeeded." (Dreyfuss: Designing for People, 1974)

Glossary

AJAX	Asynchronous JavaScript + XML
PULSE	Page views Uptime Latency Seven-day active users Earnings
UAP	User Activity Patterns
UX	User Experience
UI	User Interface
OLAP	Online analytical processing
HEART	Happiness, Engagement, Adoption, Retention, Task success
CART	Classification And Regression Tree
ISO	the International Organization for Standardization

Contents

1 Introduction

User Experience (UX) has become one of the key points in contemporary software design and development processes. It has evolved greatly in resent years and resulted in widely spread user-friendly solutions which satisfy a great variety of people all over the world. Among the web applications these are Google products, YouTube, famous social network solutions and such. All these software solutions involve so called "human-centred design". Such design approaches three aspects of the human point of view - objective, social and subjective, as mentioned by K. Battarbee. [5] All three are then addressed in engineering process and structure, form and functionality of the product. The User Experience is turning the objective from pure usability to the feelings the product produces. The word *experience* motivates developers and designers to address such used-to-be-uncomfortable words for the technological world as delight, challenge, satisfaction of the user, aesthetics of the product.

Interaction with the software causes user and the systems to co-exist and co-evolve. "Using the system changes the users, and as they change they will use the system in new ways", states M.F. Costabile et al. in their study on *End-User development* approach [11].

There is a great variety of approaches on how to involve user into the design and development process: questionnaires and feedback collecting can be used in software enhancement process, user stories may be collected for the designing and architecture of the application, user usage of the product can be observed. However, all the above methods require close user-developers interaction, which is not always possible time- and location-wise.

However, there is certain approach which allows involving user in the design and development of the application without the need of direct interaction with the user and without the demand of any time investments from the users. This approach is known as User Activity Patterns (UAP) analysis. Activity describes the flow of user's actions towards specified goal, while patterns mean step-by-step observable activity of the user in interaction with the software. As such, UAP expose the attention of the users and are representing the UX flow during the task performance, states Cole M. et al. [9]

As a tool for the User Experience assessment without the direct interaction with one, UAP is a suitable solution for the UX analysis and enhancement for the companies working with software solutions presented for a large variety of users across the globe. As the outcome of the UAP analysis can be overwhelming and present significant amount of variable data, there is a need for structured solutions on how to comprehend the results.

In 2010 HEART framework has been presented by the Google, which offered to evaluate the UX by the Happiness, Engagement, Adoption, Retention and Task success factors. As well described by Flaounas I. et al. [14], this framework not only helps to process the user experience evaluation results, but as well helps to monitor the quality and direct the modifications to be done in the future.

As an illustration for the need of the software solutions company to be aware of the UX while making use of the User Activity Patterns, the research focuses on the Raisoft software usage, which is studied through analysis of the client-to-server interaction logs. The objective of the research is to present the possibility of achieving high quality software solutions through the awareness of the user activity and experience with the product through the analysis of the widely used logging systems data. The focus of the study is to also show that the questions and answers for the project enhancement UX-wise can be easily achieved with the existing tools, such as application logs and basic scripting.

The scope of the given study includes short definition of the UX, UAP and their analysis techniques, presents applicable solutions of the given instruments for the study case and touches the HEART framework, as a tool for accessing the outcome of the UX study. The empirical part focuses on single user group, defined by location, and studies the log files for a limited period of time of one month. The patterns of interest are also limited to a few significant ones, in order to illustrate the tendencies rather than presenting a comprehensive view of the application usage. The limitations are inevitable as the software in question is large as well as the user scope, spread globally and over time.

The study though can be scaled up to cover more areas of the software and adjusted to represent variable user groups. The need for dynamic solutions for such analysis is also realised as one of the directions the further work can evolve into.

2 Understanding user

Everything in this world needs a starting point. So to say, the universal rule of beginning of all existence formed such way of thinking and affected all the processes we deal with. Also, engineering processes, industrial processes including as well software development processes do need a well-defined starting point, place of origin to start from and, if understanding it deeper, to be based on. Carpentier and Lambert [7] mention the words of a famous designer Bill Moggridge who said, "Engineers start with technology and look for a use for it; business people start with a business proposition and then look for the technology and the people. Designers start with people, coming towards a solution from the point of view of people."

As we can see, the software existence can start from or be based on three different corner-stones. The perspective then changes based on the point of origin, producing a different experience with the product as well a different existence of it (including "life span" of the application and its success). Nowadays, with the growing awareness of the User Experience impact onto the product vitals, it is needless to say that the user centred design is the correct way of achieving application success in a long run.

2.1 Usefulness, ease of use and user acceptance

According to Davis F. D. [12], user acceptance is resting on two corner-stones - usefulness and ease of use. He defines those as perceived measures, stressing the point of subjective user approach to these two. Usefulness or, better to say, perceived usefulness is basically the belief of the user in the application success in helping them performing their tasks better. It also defines some level or degree of readiness of the user to take the tool into use.

On the other hand, perceived ease of use would show what "overweights" on the comparison scale - how useful the tool is or how hard it would be to learn and/or to use it. Thus, if the perception of need to use a software to complete the job and possibility to achieve more with smaller effort on learning/using the product will be of high level, the application will be accepted by the user. As the same study shows, the biggest show stopper for the end user is often an effort. The opportunity to put less or no effort while achieving the goal, is always more beneficial despite of the other facts on the product like it's modern look and feel, highly efficient performance or similar qualities often considered by the development as the key points of the successful product design.

As A. Bandura [3] says in the research on human psychology and self-efficacy, self-efficacy shapes human behaviour. It can either help to or prevent from doing some tasks. It also has an effect on the personal choice towards or away from some products, tools, activities. If a man is not happy with own performance or does not believe in his/her capabilities in using some tools, he/she will avoid them if possible or be unhappy and unsatisfied with them if impossible to avoid, as they will produce great deal of stress for the person.

Thus, while designing, developing, enhancing the software, it is important to consider the user groups' capabilities, what they are strong at, how is it possible to make the program assist to the users' professional strengths and also deploy those strengths in the program use. As mentioned in the publication [3], the person becomes more attentive if he/she experience success in controlling the process. To achieve the perception of control, the software should be designed "around" the user, answer his/her needs, strengths.

2.2 Challenges of being a user

In the process of adapting a user-centred approach, it is important to understand the challenges customers meet while being asked to perform their daily work tasks with help of yet another software solution. Usually, the user groups are not educated in the Information technology field, have other complicated tasks on their daily work calendar and are preoccupied with achieving the same or better results with help of the new or enhanced tools. Thus, when asked to use new software solution, the users, while performing their daily often non-trivial tasks, find the software use a destruction rather than a help, notes E. Wagner et al. [40].

Often the challenge of using the software lays in the high level of cognitive effort which is required from the user.

2.3 Beautiful user

The term Beautiful user has been presented in *Beautiful users: Designing for people by T. Carpentier and T. Lambert*[7] work, which focuses on revealing the need and beauty in user role in the design and development process of products and services. As it is well described in the paper, users have been assigned lots of variable roles in the process of designing the product. They were considered as normative types (looking for commonalities), diverse types (looking for differences), consumers (observation targets), those who are to be measured and manipulated, and finally as dynamic partners.

In the process of user centred design and development, users play a role of an outside observer, seeing the bigger picture, reacting to the product and analysing the result while experiencing and interacting with the software. As such, users add to the view of the software greatly when instead of knowing the parts, they view the product as a whole, as B. Moggridge - one of the pioneers of interaction design - notes [25].

Users are also being a moving force of the software development. Especially the new user make the environment evolve, as they demand from the developers to adapt to their needs and use habits [11]. Software designers and developers simply cannot ignore the fact, that their product success is greately dependent on the knowledge transfer from the users to the design and development team [18].

As mentioned by D.W. Conrath et al. [10], today's situation with the user tolerance towards the drawbacks of the business process or product design has been lifted up, as the users are much more knowledgeable and are not satisfied with the complex features, time-consuming solutions, unnecessary complexity and such. Their involvement in the design makes them both - more demanding but also responsible for the solution. And, as the software market grows exponentially, the user requirements become more distinct or precise. While having lots of ready general solutions for the basic tasks, customers often order specific software application to answer their needs with high level of precision. Thus, working with today's user is yet another challenge, but also the place where the development team can get majority of what they need for succeeding with the solution.

2.4 User based design

Finally, with a time of trials and errors, the product developers world-wide are coming to the understanding that *designing for people* must be correlated with *designing with people* [7]. One can argue that the user driven design and development is already in general practice. But, if we consider all the facts, it will show us that often short-term interests overcome the desire to act on user request. As such, short term interests are short term financial gain, intent of the developers or designers to express themselves, an opportunity to reduce manufacturing or production costs (finding cheaper solutions or materials), or simple way of habit (it is always psychologically easier to follow the well known pattern or path). As well, there is an argument of user inability to explain the needs and tasks in technical terms due to the luck of knowledge of the software field, and development team inability to relate to the user stories due to the lack of knowledge on the field users are operating in [18].

Luckily, the contemporary approach to the product design and development is turning to avoid such short-term factors and user-developers knowledge-base gaps and focus on manufacturing the high quality solutions, which is only possible with the user centred design. As it has been pointed out by one of the *Museum of Modern Art's 1988* projects curator Cara McCarty, "people are "disabled" by obstacles in the environment; once those obstacles are removed, disability falls away" [7]. The same idea is applicable to the digital world. When the developers and designers look at the application not from the developer's point of view (looking at its functionality and ability to perform some tasks), but rather from the user point of view (understanding in which aspects an application causes the user "disability"), the design and development process will produce more user friendly solutions and will gather larger amount of satisfied user groups. "It is imperative to involve the user in the design process, for the objective is to develop aids that will make maximum use of a person's abilities...", says McCarty.

Designing with the user can be done in two ways - applying agile methods and using User centred design technique. They differ by the process itself and the places of user involvement, but both methodologies do place users in the center of the process, considering their knowledge and point of view. Both methods are well presented by D.Fox et al. [15].

According to the publication [15], the agile methods approach place customer satisfaction at the top of the priorities list. That results in the process of short term feature developments, implementation and deployment. The key parameters are user request, short implementation and fast results. In the agile development, the user story is the corner stone for developers, they are studied, prioritised and acted on. Such design helps to avoid the situation, when the user is not satisfied with the product even though it was made on his request initially. As it is often impossible to predict all the key points of the software and define the application structure perfectly on the initial stage of design.

The methodology is widely used by famous or not famous companies, and is proven to be a working solution towards enhanced User Experience. One of the most famous agile companies is Apple, as Scrum master Lynda Menge claims [24], it is taken the advantage of agile approach towards not only the development process but all the other processes in the company, such as training and learning processes. Agile approach has not only produced better User Experience and made users generally satisfied with the product, but also enhanced productivity of the teams in the development.

Another approach - User centred design - is on the other hand based on the thorough investigation of the user needs and long preliminary stage of preparation with careful consideration of the user needs and wants. According to D. Fox [15] this stage usually involves three steps - deep study of the user tasks, environment and users themselves; evaluation of the usability and usage of the product, understanding how easy or difficult it is to use it, understand it, learn it; fixing the issues, pointed out by the users during the second step.

Examples of the products developed with User centred design techniques include Duolingo, Trello, Airbnb, Yelp, Instacart and other well used applications as described by N. Asad [2]. As can be seen from the Duolingo example, the language learning tool is designed to be simple, with no confusions on the navigation or functionality of each page and view, it serves all the users despite their language, skills, origin. It is simple, yet powerful.

As the need to involve the user into the designing and development process cannot always be achieved directly, some other solutions can be found to achieve this goal. One of the well-known ways is in collecting user feedback and acting on it. Another approach is to study user activity within the application and study the patterns of user actions, which produce even more detailed and accurate results on the user - product interaction and with the correct analysis helps to involve the user in the design completely indirect. Studying User Activity Patterns is an effective way to realize software "disability" triggers, parts of the application which do not simplify the user task solving but rather slowing the process down causing dissatisfaction and sometimes frustration.

3 From User Activity to User Experience

Relations between software development team and software end users is usually remote and often in order to get some message across from one side to another it needs to travel through a chain of company hierarchy. It is obvious that on that way, some specific information is hard to deliver, unlike if it would be done in face-to-face communication. These types of information are usually non-verbal, connected to impression and experience, feelings, satisfaction. Often it could be considered as an unneeded sort of data in productive discussion, but as the last years has shown, these sorts of information unified now under the term User Experience (UX) is greatly undervalued and highly needed type of data for developers and development team to consider. Relying on UX can make the software a successful product with growing amount of dedicated users, on the other hand, focusing only on functionality might break the relationships between end users and the product developers, in worst case scenario, destroying the market for the software.

The need to consider and apply the UX awareness to the design and development process of the software for ensuring the product success, leads to the understanding of the need to somehow define and measure the UX in current usage of web applications through realisation of the user's side of the coin as well as the development team's. Awareness of the diversity of the user groups using specific software shows the need to approach User Experience questions through the prism of generalisation, presented in classification, unification and patterns.

The need to simply adopt the work done to the rest of the application, data scope, user groups, produces the understanding of the need to also by chance think about the future possibilities of automation of the process of user activity study and analysis and User Experience monitoring on a general bases.

3.1 User Activity Patterns

User activity patterns during information search study by Cole M. et al.[9] has defined User activity and User Activity Patterns in a well-described manner.

According to the article [9], User activity describes constant and current user's actions flow while he or she is working on solving some task or achieving some goal. Such so called moment-to-moment activity can be monitored in e.g. different users, sessions or circumstances and as a result a pattern of actions can be recognised, well defined actions' pattern in interaction of the average user with some typical information object.

Based on the same study [9], User Activity Patterns (UAP) can be defined as user action sequences which describe the user's information behaviour in some particular environment. Such patterns reflect both the plan how to carry out some intent and the intent itself. So, to carry out some task, the user goes through several behaviour steps consisting of intention-plan pairs. Thus, it is obvious that the user behaviour will be directly dependent on the task itself and its properties. As such, the complexity of the task will be one of the properties which will affect the user behaviour greatly. The complexity does not only define the task itself, but as well the user, for example in his or her usefulness in the process, in the way how relevant is the task towards his or her position in the process, what is the user's experience with the task, his or her stay time in the project or specific part of it.

Elaborating on the user to task relevance and his or her expertise, it is worth mentioning the need to identify and address the differences in user groups when defining the user activity patterns, that is crucial to the success of application design that can meet high UX standards, as *Managerial use and emerging norms: effects of activity patterns on software design and deployment* by Grudin J. [16] also states. Differences in user groups can be seen on all possible levels of interaction with the software: software usage, task understanding and efficiency in its completion, cognition, knowledge, preferences in actions, look and feel of the software. Overall, peoples work style is different, so their ways to achieve the same goal would always vary. It is worth mentioning that some differences are also highly specific and are not easy to address, e.g. existence of color blind users. Such differences cannot be overcome but the level of their affect on user activity patterns should be realized and approached. Also, the organization will always have a long list of variable users, generalization on their activity can be also found when learning the differences in their UAP. Such generalization can help to narrow down the patterns to a few which would cover most of the members of the given process, thus this will help to provide unified solutions, which would recognize usable parts of the software, highlight unused features and help finding the needed bridges between sequentially used sets.

Cole M. et al. [9] mentions that the observations on the user activity can be done from a variety of view points, such as on an application level (page to page, window to window switching), page level (dwell time in the page, what content it has, which content is used, in what order), and action level (clicks, scrolls, navigation needs). Most of these are usually easily available for observation as they are usually recorded in the custom user log files or can be recognized with correctly equipped web browser set of tools. Finding the User Activity Patterns helps to place attention on specific parts of the application, observe relevance and usefulness of variable parts of it, find areas of specific users' attention, and this leads to understanding of the user experience in the session flow.

Thus, the center of attention in studying and recognizing the UAP is focus on user-to-data interactions, that helps to approach the system not in sense of e.g. documents hierarchy or relevance, but rather in sense of usability of variable parts of information or features to the end user.

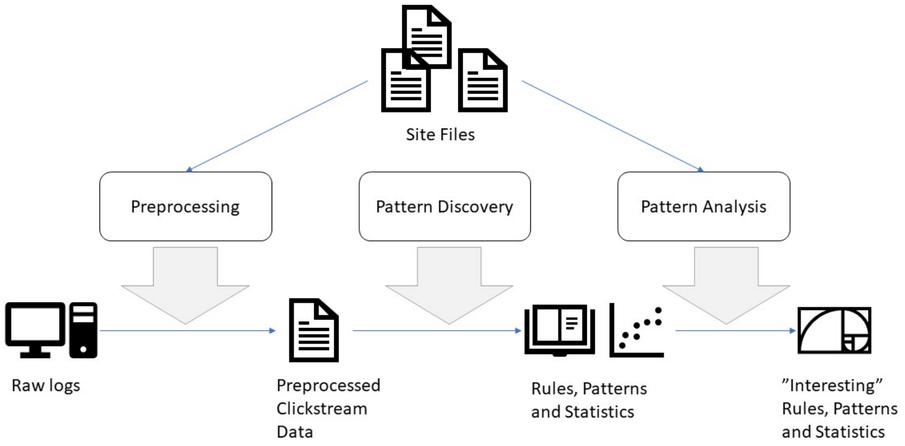

Site Files
Preprocessing
Pattern Discovery
Pattern Analysis
Raw logs
Preprocessed Clickstream Data
Rules, Patterns and Statistics
"Interesting" Rules, Patterns and Statistics

Figure 3.1: High level web usage mining process, based on J. Srivastava et al.[37]

According to Srivastava et al. [37], UAP recognition algorithms and methods are mentioned to be based on a variety of data processing fields at the same time, they take use of data mining techniques, statistical analysis, involve pattern recognition and also machine learning. When studying the Web based application, several pattern recognition algorithms can be mentioned. Most of them are well applied to the Web based search, Web application navigation and Web usage.

User activity patterns recognition and analysis are defining steps of web usage mining process (see Figure 3.1). The process includes following activities: firstly recognizing the log data, discovering the patterns and finally analysing the patterns. But some studies are also proposing to include data preprocessing as a first step of user activity pattern processing, for example this step is crucial in dynamic web activity processing tools, which are not set for doing any preliminary check and adjustment of income data by hand, but rely on automatic selection of the needed information [4].

3.1.1 Pattern Discovery

Over the time, a variety of methods has been used to search the raw user activity data for patterns. According to Srivastava et al. [37], among those, methods applicable to Web domain are: statistical analysis, association rules generation, clustering, classification, sequence recognition and dependency modelling. It is important to mention that the methods listed above and described below in more details are not overall but rather specific, giving the view on the similar user behaviour from different "points of origin". A good example would be seeing that any basket-type analysis like association rules generation will not realize the order how items are selected, while sequential analysis will focus on the order and will not recognize the patterns with the same steps but different order as similar ones. The approaches can be described as following:

- Statistical Analysis - such an approach provides a variety of statistical measures, which can be very helpful when searching for commonalities, says Srivastava et al. [37]. Such measures include median, mean, frequency, of e.g. views, time, clicks, navigation paths. There exist Web tools, so called traffic analysis tools, which can be used to perform such analysis. The tools can be configured to provide report on the Web activity periodically, informing the tester about the view time of the page, path lengths, possibly also errors on a low level, such as unauthorized logins, non valid URIs and such. In-spite of the fact that such analysis will not provide any deep knowledge on the user session and activity, the information received from it will be still useful in sense of UX, when considering the improvements of Web application performance. Statistical analysis are often used in learning of so called stationary probabilities of the application usage, according to P. Tonella [39]. An example of such probabilities are pages usage or accessibility, navigation models, they are defining how the application is navigated through and help to design placement of the data parts and features in sequence with the needs of the user.

- Association Rules - this method is very useful in Web application user activity patterns discovery as it works on relating the pages or features which are accessed in sequence in the same session. One of the known tool used in Web Usage Mining context is the Apriori algorithm and its variants, which shows the correlation between users who accessed certain page or page sequence. According to Srivastava et al. [37], this type of pattern discovery can help to solve two important tasks: redesign the Web application or site to apply useful structure based on user activity, making it easier for users to perform the same sequence of actions with fewer number of clicks or navigation steps; preload the needed documents based on the pattern discovery prior user clicks, to ensure better performance. A trivial yet very descriptive example of the method usage is a market basket. As described by A.Rajak [32], market basket analysis are often done in supermarkets, when the data, collected by the bar-code scanners, that includes shopping list per customer, is processed in order to define which products has been bought in groups. Processing such data provides managers with the idea how to cross-sell, promote, design catalogues, segment customers and simply locate the goods within the shopping area logically.

- Clustering - another grouping method, which can work either on users or on pages. As Srivastava et al. [37] states, clustering of users help to see the same activity patterns in groups of users and define the user types, which can help to predict the usability sequences and patterns of the Web application based on the user types using services most. As such, when starting to work with the new organisation, the Web application providers can predict the needs for certain application features prior the use of the application even begins. Page-wise the clusters of pages used in the same session or when performing some specific activity can be discovered. Unlike in Association Rules discovery, the clusters will not take into account the sequence but the set itself notes Srivastava et al. [37]. This will help to define pages or features of the same type, which can be used in creating better look and feel with needed links and bridges between tools and features of the same cluster. This methods' outcomes can also be used when developing application usage assistance tools. S. Nirkhi et al. [27] draws the conclusion on clustering being one of the main techniques for search engines working with very large datasets. Services like Yahoo, Ask and Google use spiders programs to organize and manage databases. They crawl through the web, finding pages, then sending them for indexing and storing in the search engine files for searching by keyword or other available parameters. As the web grows, clustering is used in web search engines to either identify the needed pages or to present the result of the search to the user. With help of clusters, the engine may present up to a thousand results to the user view in a matter of seconds, and the user will be able to browse through them more efficiently, while without the technique the top limit of the documents will sum up to a hundred, says S. Nirkhi [27].

- Classification - this method requires predefined set of classes, then the data items can be mapped to one or more of them, notes Srivastava et al. [37]. This is also useful when trying to split the users into groups for discovering the similarities and differences of their usage of the application. When creating the classes, it is required to gain a deep understanding of the system properties which are to be highlighted, thus the classes will categorise the set of users in a sensible and useful way for this application development and design. The known classifiers for such tools are k-nearest neighbour, naive Bayesian, decision tree, and Support Vector Machines. The best illustration for the algorithm usage is the email spam filter. As the A. Sharma et al. [36] research presents, there are several useful types of classification algorithms that are used in spam recognition, for example CART (Classification And Regression Tree), which is using historical data set with predefined classes to make observation and to build the decision-tree.

- Sequential Patterns - technique of finding the items sets which follow one another or specific item which follows specific set, according to Srivastava et al. [37]. This method can be used to predict the possible set of steps the user will take when performing similar task, which would be useful knowledge in the Web application optimisation design process. As it is stated in the publication by M. Joshi et al [20], sequential nature has been discovered in many different sets of data - scientific experiments results, physical systems monitoring, IT networks, supermarket transactions and such. That is the reason why sequential patterns is the methodology applied widely in data mining. This method is described in more details in Section 3.3.2.

- Dependency Modelling - this analysis technique is meant to define dependencies of great significance which tend to exist between some variables. Srivastava et al. [37] defines this approach to be useful when, for example, discovering the user activity stages or actions, their browsing behaviours and similar activities. As such, learning techniques as Bayesian Belief Networks and Hidden Markov's Model are applicable here. Plus the fact, that Web usage patterns modelling is providing the needed framework for user behavior analysis, it also can assist in prediction of the resource consumption, define the areas in the application which need to be optimized, lucking features and find possible application usage bottlenecks, the above mentioned publication says [37]. Dependency recognition techniques are widely applied in artificial intelligence, because as J. Pearl justifies [30], these models remind human models of cognition and decision making in the situations of uncertainty. The technique, for example Bayesian network, "captures patterns of reasoning" [30] by recognising forward and backward dependency between certain items, actions, phenomenon.

3.1.2 Pattern Analysis

According to Sreivastava et al. [37], the final step in UAP analysis is to perform the actual analysis of the collected patterns by filtering out all the rules which present no interest to the performed study but has been discovered during the stage of Pattern Discovery. The method which is going to be used here is completely dependent on the type of Web application the investigation is performed on. Usual way of doing the analysis would be to use SQL or another type of knowledge query mechanism. Other methods of analysis would include some type of visualization techniques, which could help to graphically or visually show the content filtered in certain way, here basic highlight can be used, "find in page" filtering applied or OLAP (Online analytical processing) operations performed on the data after it has been loaded into a data cube. Yet one more technique would include processing the data through some script, specifically designed for the data set and outcome requirements defined by the researcher.

As mentioned in *Dynamic Personalization of Web Sites Without User Intervention* [4], Pattern Analysis step is meant for choosing the patterns of interest and discarding the rest. As such, this step is highly specific to the application in question, user group, and the needs, defined by the research.

3.1.3 Similar studies

As a justification for the UAP and logs data applicability for analysis and their usefulness for developing better services for user without the need to distract the user with questionnaires, feedback gathering or other data collection mechanisms, a few similar studies can be presented.

One of the studies [42] has shown a great possibility to use logging system for gathering time activity data of the user group and developing better safety mechanisms for them . As seen from the experiment performed in the study by C. Wu et al.[42], the knowledge-base has been built without the need to collect any extra data from the users in questing either than accessing logs from their mobile devices. The study has focused on acquiring the information on people's time-activity patterns, to recognise the time and period they spend outdoors and indoors, for estimating the possible levels of exposure of population to the air pollutant.

As described in the User Activity Patterns 3.1 section, the study also went through the same stages of:

- Gathering the data - mobile usage HTTP requests log of G2 and G3 cellular network has been collected
- Preprocessing the data - instead of using one step at a time, the steps have been gathered into clusters, defined as Continuous actions, and viewed as a single step in further study
- Pattern discovery - the methods have been developed to study the indoor and outdoor activity times, house and office presence and also distinguish more precise indoor activities
- Pattern analysis - the time outdoor has been calculated through recognition of the shortest way between home and office.

The results has been compared to the questionnaire collected from the same group of users prior to this study, showing the accuracy of this approach. It is also worth mentioning, that the study has been performed on a very large group, and the success of this experiment proves usability of UAP study from logs to be successful and accurate way of learning. From the actual benefits of such method, the research group [42] has mentioned three: phone logs being more accurate than data collected from the memory, mobile web logs covered about five million of people while direct user approach could never be that broad, finally studying the logs is cost and time-effective solution for both researchers and users of the product.

Another study put even more challenging task as a goal - producing constant User Experience enhancements based on the user activity and UAP recognition. The study presented a solution for *Dynamic Personalization of Web Sites Without User Intervention* [4]. The experience went in a similar manner to the given research, by extracting the knowledge on the UAP from the logs, preprocessing the data, discovering and analysing the patterns, applying the patterns of interest to produce better user interface on a custom per-user level.

Unlike in other applications, which focus on finding the similarities in user actions and defining the general UX scope, the application presented in this study, it focuses on user profiling [4]. Still, the same approach to pattern analysis gives suitable results for achieving the goal.

The study suggests using a system called SUGGEST to achieve the dynamic page suggestion based on the user preferences [4]. The main pattern discovery algorithm used is clustering. The logs are processed to produce "session clusters", pages accessed more often and closely together are grouped into clusters, the ones which are met more often (above the set threshold) are then used to develop a list of suggestions, and those which are below the threshold level are discarded.

As can be seen from the examples above, log processing for patterns recognition is rapid, effective and simple yet powerful way to acquire the needed knowledge on the user activity and experience.

3.2 User Experience

3.2.1 Defining UX

It is now obvious that defining User Experience (UX) will be challenging, as it describes dynamic process, happens to be highly context-dependent and is subjective to some extent. According to Law E. et al. [22], it can be defined from individual site rather than as social and shows that individual's interaction with some object of interest, e.g. service, system or product.

As stated in the *Understanding, Scoping and Defining User Experience: A Survey Approach* [22], it has become obvious that traditionally developed framework with its primary focus on the cognition and performance has its drawbacks in recognition of the non-utilitarian experiences in interactions between the human and the machine. Thus, the need for such a field in computer science as user experience was found necessary, as UX focuses on user sensation, affect, satisfaction. As such, UX becomes a desirable feature of every software application developed for interaction with the user, but it is something which is to remain open for rethinking, redefining, for being debatable and modifiable.

There are several challenges in defining the UX:

- Key concepts of UX are somewhat unclear, subjective and non-static. They include aesthetic, emotional, experiential variables. Definition of the set of key values also may vary depending on the observer subjective decision whether some concept needs or needs not to define someone's experience with the subject of interest, states Law E. et al. [22].

- Analysis units are malleable, says Law E. et al. [22], they can range from one to many user's interaction aspects and include from one to many users, overview all the services provided by the company or application or focus just on single product or feature. It is important to define the ranges of interest clearly.

- Complicated or fragmented landscape of the UX phenomena becomes obvious due to diversity of the theoretical models that try to explain the concept of user experience. All these models tend to focus on a different set of points, including experience, beauty or affect, emotional response, pleasure on one side or pragmatism, value and quality on the other, notes Law E. et al. [22].

- Diversity of the UX, presented by A.H. Allam [1], adds to the complexity. The diversity is presented in such aspects as individual experience, product type, situation in which the product is used and time user spent with the product.

Thus, different sources offer a set of definitions presented below, which in total describe the UX concept most fully. User Experience is defined as:

- "user's perceptions and responses that result from the use and/or anticipated use of a system, product or service" ISO 9241-210 [28]

- "All aspects of the end-user's interaction with the company. Its services and its products." D. Norman [29]

- "A consequence of a user's internal state (predispositions, expectations, needs, motivation, mood, etc.) the characteristics of the designed system (e.g. complexity, purpose, usability, functionality, etc.) and the context (or the environment) within which the interaction occurs (e.g. organisational/social setting, meaningfulness of the activity, voluntariness of use, etc.)" Hassenzahl and Tractinsky [17]

- "The entire set of affects that is elicited by the interaction between a user and a product including the degree to which all our senses are gratified (aesthetic experience) the meanings we attach to the product (experience of meaning) and the feelings and emotions that are elicited (emotional experience)." P. Desmet et al. [13]

- "The value derived from interaction(s) [or anticipated interaction(s)] with a product or service and the supporting cast in the context of use (e.g. time, location, and user disposition)." Law E. et al. [22]

- "The quality of experience a person has when interacting with a specific design." Law E. et al. [22]

These definitions conclude the set of ideas stating with the main idea behind UX being a subjective experience of the user with the dynamical and changing environment of the e.g. software, and all the cumulative parts in the UX "formula" as variables - user, context, benefits and drawbacks of the system.

Law E. et al. in their study [22] propose a good visual representation of the UX vs other experiences, see Figure 3.2.

3.2.2 Measuring UX

In the latest years with rapid growth of Web based services, the need for analysis of the web site metrics and tracking of the user patterns and experience has triggered the demand for some tools to perform such measurements. Thus, a large amount of solutions in this area has appeared. As everything in the Web world, the solutions presented in the marked are both free-of-charge and commercial, based on variable languages, developed for specific or general purpose and platforms.

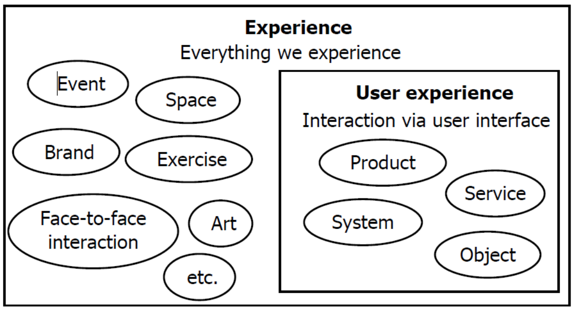


Figure 3.2: UX in relation to other experiences that we can study [22]

Large research by Hussain J. et al. [19] classifies the means for UX evaluation in three main categories:

- self-reported measurements
- observational measurements
- psychological measurements

In the first approach, the user group is filling up a questionnaire, attends an interview or similar to openly express their emotions about the subject, in second, the data on the UX is collected by observers - specialists or software - to recognise user interaction with the subject, in third, sensors are attached to the users to collect the reactions of the user [19]. The most traditional and well used methods are self-reported and observational. Self-reported has been in a favour until the recent times, when user groups of variable applications and services grew up exponentially, and questionnaires or dialogues got to be impossible to perform on the given number of people. Thus, the optimal solution for the Web based applications UX research are observational measurements.

J Hussain et al. [19] argues that with the help of observational methods it is possible to resolve problems existing with the other two methodologies, as the method can be done in a way, that the user group will be unaware of the ongoing observation and, as a result, will act more natural, perform tasks as usual and won't be distracted by the need to adjust to the form of questionnaire or open discussion. But still, the method won't produce comprehensive results concerning user feelings or emotional state, unless the observer will have needed expertise to correctly translate the behaviour. Basically, emotional side of the UX has been and always will be a difficult area to define both by the user and by the researchers, as those might be also hard to express, difficult to recognise and sometimes impossible to read. One poor solution for this would include biometric readers.

Rodden K. et al. [35] states that the existing and evolving solutions for the analysis metrics are still not completely effective and applicable in every situation, as they can be seen as too general, when applied to a specific product in research of particular quality of it. Also, the data amount coming with the Web usage process is so large, that it is obviously difficult to process it without initial clarification and the necessity to scale it down to some defined scope, which in the case of every software will again differ. Finally, there is a need to scale down the set of actions which are to be taken in every particular case, to produce the needed result and to answer only the set of questions of this particular measurement.

To address the last described issue, experts suggest to focus on a defined amount of user goals or key business goals and then use specific metrics see how these goals are achieved and track the process towards these goals. But, as Kazman and Bass state [21], the opportunity of accuring clear business goals has always been fine, as they tend to exist in a variety of forms and abstraction levels.

The study described by Rodden K. et al. [35] mentions that this approach is clear and logical, but achieving such sunny day scenario is not always possible, as the service providing teams usually get to work with the product team, for whom defining the goals and clearly describe them is not always easy. To help with this process, it would be useful to support the metrics definition with the findings, that can be provided using other sources of information, such as field studies, usability studies.

In different publications e.g. by Rodden K. et al. [35], by I. Flaounas and A. Kokkinaki [14], the process of User Experience evaluation is based on top of three main keystones: goals, signals, metrics realization, where steps are to be present in this particular order.

According to the proposed solution [35], identifying the goals in this case would mean answering the questions: what are the tasks that users are in need to accomplish and what would be achieved through the redesign. Identifying the goals must be the initial step and metrics will be unusable unless they do have the set of goals well defined. Because the metrics in itself is something that helps to see whether the user can progress towards the goal while performing tasks in the software environment and how easy, comfortable and straight-forward his progress is.

For example, Rodden K. et al. [35] team proposes paying attention to following tips when working on goals:

- Goals can be different for different members of the team. They can even cause disagreements. The best way is to collect all the proposed goals, and then work through them towards the common goal. After reaching the consensus, go with the chosen metrics.
- Goals can differ based on the segment of the project in question, goals for the project in total may vary from the ones for a feature or module.
- When brainstorming on goals, it is needless to spend time thinking whether they will have corresponding signals and metrics and what they are.

The question "how easy, comfortable and straight-forward" the process of achieving the goals is leads to another step defined in the process of UX measurement, as mentioned in e.g. Rodden K. et al. [35], which is identifying the signals. The questions asked in this step are: what actions indicate that the goal has been met and what kind of feelings will indicate the failure or success.

When working on signals, Rodden K. et al. [35] team proposes paying attention to the following tips:

- The signals of choice must be very specific for the defined goal. They should be sensitive enough to the changes towards (or away from) the goal, but react *only* to the UX changes in relation to the set goal and nothing else
- when choosing signals, it is needed to keep in mind that it is often easier to check for failure rather than success. Prove by contradiction is also a strong argument.

After realizing goals and signals indicating that the process towards these goals are positive or negative, the unique for this product and process metrics can be finally developed.

Realizing the following aspects based on Rodden K. et al. [35] is important:

- It is not always good to collect general data and present statistics on the user group as a whole, as the amount of users can grow rapidly. It is also a good practice to collect statistical data per user, sometimes even more useful.

- When working with user logs (web logs, server logs), it is essential to preprocess the dataset, removing unneeded records, ensuring careful login of all the needed activities.

- It is a good practice to have chance on comparing own project to the existing ones. Sometimes it will require additional metrics, which are applicable to the other projects in question.

One common type of metrics which are widely used for measuring the product's technical sides and monitor the general condition of a product over time is called PULSE, and is well described by Rodden K. et al. in *Measuring the user experience on a large scale: user-centered metrics for web applications* [35]. The abbreviation for these metrics is PULSE, which comes from the main aspects which are measured in these type of tests: Page views, Uptime, Latency, Seven-day active users and Earnings.

Even though these metrics are somewhat descriptive and applicable in sense of User Experience and do give some indication on user satisfaction or opposite with the product (for example, when measuring the latency, low latency can be assumed as a positive factor for UX), these are still too general and not UX specific. For example, it is difficult to measure the progress or regress of the UX with the User Interface changes, states Rodden K. et al. [35]. It can even result in false assumptions, as for example page views of high number might indicate users' interest in the page content or as well be a signal of malfunction of the UI, where the user is directed to the page which he/she is not interested in, but false links or poor design make it difficult for the user to escape the step of returning to this particular page. As such, false assumptions may lead to UI development to go in the direction, which will seemly bring the success and rise the revenue in short term, but would draw the users away in cause drop in interest in the feature long-term.

The stated problems indicated the need for development of User Experience oriented metrics on the base of existing ones, which would specifically serve the need of measuring the UX and defining the impact of the updates in UI to the UX. Rodden K. et al. in *Measuring the user experience on a large scale: user-centered metrics for web applications* [35] proposed the following solution called HEART metrics. HEART stands for Happiness, Engagement, Adoption, Retention, and Task success.

The "Happiness" term stands for anything what can be concluded in concept of attitude towards the product, holding the answers to such subjective questions as: how easy is this product to use, how visually attractive is it, does the user "like" it, is the user satisfied with it and is the user ready/willing to recommend this product to others. This part of the measurement can be done with surveys and questionnaires.

The "Engagement" step is something which is also measured by PULSE, and includes involvement of the user in usage of the product. It considers all: how frequent are the interactions with the software, what is the intensity and how deep are the sessions in sense of features usages, time spent on the tasks, completions of the tasks. g the UX and defining the impact of the updates in UI to the UX. Rodden K. et al. [35] mentions the importance of performing this step of measurements on user level rather than paying attention to the overall numbers.

The "Adoption" step grouped with the retention step basically show what is the user number and how steady it is. While adoption would define the number of new users who started with the software in some defined period of time, retention would show, which of these stay with the software or some product after specific period. Thus, adoption would show the quantity while retention would indicate sort of quality of the user number.

Finally, in the step of "Task Success", the User Experience metrics are applied, such as efficiency, effectiveness and error rate. Efficiency and effectiveness would indicate the task completion time and percentage respectively, while error rate would indicate the drawbacks in the task completion process. These measurement can only be trivial in case of particular path of steps existing in the task process, otherwise it could be user-dependent and task-dependent yet again, as each user might take his own unpredictable way in achieving the same goal.

3.3 Selected metrics for data processing

The choice of the selected data processing methods was based on the logging data provided for the given study. Due to the data being collected from a large amount of users over 28 days and recorded sequentially into the files, the choice went for statistical analysis and sequential patterns analysis.

3.3.1 Studying patterns using Statistical analysis

When dealing with large data scopes, which have some defined tendencies and repeated events or records, it is obvious that basic statistical analysis will provide sufficient information on the recorded process. The methods in question are:

- calculating averages (mean)
- finding maximum and minimum, ordering by occurrence
- finding presence or absence in the set

The outcome of the statistical data calculations is presented in a separate file, and even not being the primary mean for pattern analyses, statistics indicates the tendencies and produces needed additional data.

As stated in *Mathematical statistics and data analysis* [33], mean, median and other methods of so called measures of location are widely used as a tool for producing reliable estimate or summary of the quantity. As such, average calculation, which is described in the well-known formula:

$$\frac{1}{n}\sum_{i=1}^{n} xi \tag{3.1}$$

is used in showing the resources consumption in application usage, such as time efficiency, methods usability and session length.

According to J. Rice [33], mean calculations are highly sensitive in sense of being prone to wrong assumptions being produced if the data supplied for the calculations was not sufficient. The main danger in mean calculations is to have the outlier samples within a set. Outliers are usually error measurements, caused by some equipment malfunction or such.

Careful consideration of this fact in connection to the data in this study has proven the usability of average calculation method in given case, as the e.g. timestamps and function calls being studied cannot produce falsely high or low results. Timestamps are recorded sequentially and cannot produce outliners, as well function calls are always compared to the set of known methods, which means the result cannot run beyond the known scope.

Other statistical methods used in the study for data description is distribution of samples, which is basically an ordered arrangement of segments, minimum and maximum calculations and coverage study. Simply put, all the tools of descriptive statistics have been applied to the data in question, as the descriptive statistics is suitable for summarizing existing data sets rather than producing any conclusions on the scope beyond the existing, predicting or presenting hipotheces, as mentioned in *Descriptive and Inferential Statistics material* [23].

3.3.2 Discovering patterns using Sequential patterns method

Sequential patterns occurring with some value of frequency has been known as one of the most famous and useful practices in data mining field as both Morzy [26] and Wojciechowski [41] state in their publications on SQL manner data analysis. As the first publication notes [26], sequential patterns can be applied to such fields as medical records analyses, buying patterns discovery, telecommunication systems analyses and similar. Thus, user activity patterns in the application which deals with patients records creation, maintaining and analysing would be as well a suitable field for sequential pattern discovery practice.

Besides the discovery of patterns, which frequently occur in event sequences, there is another approach to the same techniques described in Wojciechowski publication [41], which focuses on so called episodes. This approach addresses single sequence searching for specific patterns within the sequence.

S.Thomas [38] presents clear throughout view on the sequence recognition technique. According to it, the basic items in the process are:

- a set of sequences
- a sequence as a list of transactions
- a transaction as a set of elements, e.g. user actions, marked by a timestamp
- a sequential pattern is a defined sequence
- a sequential pattern support is a number of sequences in an input dataset which contain the defined sequence
- maximum or minimum timegap is an additional quality of a sequence, which can be used if needed to define the boundaries of a sequence

And, based on the same source [38], the default way how this algorithm works is the following:

- a set of sequences is given as an input, where each sequence is actually a transactions list, the order of the transactions in the list is set by the time when transaction has occurred
- a sequential pattern is given, the one which is to be searched for in the given data

- a set of sequences is compared to the given sequential pattern, defining the amount of sequences matching it

As M. Wojciechowski states [41], the goal of the sequential pattern analyses is to discover all episodes in the given data sequences which would satisfy user-defined ordering, ordering conditions, content conditions, taxonomies and time constraints.

A thorough explanation of the sequences discovery is presented in *Fast Discovery of Sequential Patterns Using Materialized Data Mining Views* by T. Morzy [26]. According to it, the process of sequence discovery can be described using a non-empty set of literals denoted as

$$L = \{l1, l2, ..., lm\}$$

from which the sequence is constructed, or in other words, there is presented a sequence

$$X = (X1, X2, ..., Xn)$$

consisting of elements $Xi \subseteq L$, which are ordered.

Thus, if we define a set of sequences D, each of variable length, where sequences are defined as $X = (X1, X2, ..., Xn)$ and each element of the sequence X is defined by a timestamp, we can note that the sequence $Y = (Y1, Y2, ..., Ym)$ contains prior defined sequence X if there exist iterators (simply integers) $i1 < i2 < ... < im$ so that $X1 \subseteq Yi1, X2 \subseteq Yi2, ..., Xn \subseteq Yin$. And in such a way, the $Yi = (Yi1, Yi2, ..., Yim)$ sequence would be called an occurrence of sequence X in sequence Y. Such direct analyses is possible if we do not define any time constraints.

But, the time limits can be additionally applied to the search mechanism, defining, in what time the sequence should appear ($minimum$, $maximum$ timestamps), what is the allowed gap between consecutive occurrences ($min - gap$, $max - gap$) and maximal duration, which defines the time span that allows to treat a few elements following one another as a single one, it can also be defined as time tolerance ($time\ window$).

The sequential pattern mining produces two basic results: $support$ of the sequence and number of $occurences$. The first statistical measure looks for the existence of the sequence in the data set D, while the latter provides researchers with the distinct number or fraction of suitable data in D which corresponds to the given parameters, Morzy states [26].

3.3.3 Pattern analysis using scripting

The process of data analysis is a broad accumulation of a variety of stages, as described in *Scientific Data Analysis using Jython Scripting and Java* [8]. As the author mentions, it consists of at least following steps:

- Data gathering and transformation or formatting
- Data reduction to the needed volume and cleaning
- Data description with statistical analysis
- Data knowledge discovery, also known as data mining
- Data comparison with other data sets
- Data modelling - processing data with models, either numerical or analytic
- Data visualization and presentation

According to the description above, data analysis can be comprehensively approached by custom developed scripts in some widely used scripting language. Well developed script can solve all the needed issues with the processing and present the results in custom manner, as the application requires. As well script can be further developed to either serve different types of data or be executed on dynamic input. That is why scripting was the choice for pattern analysis in this study.

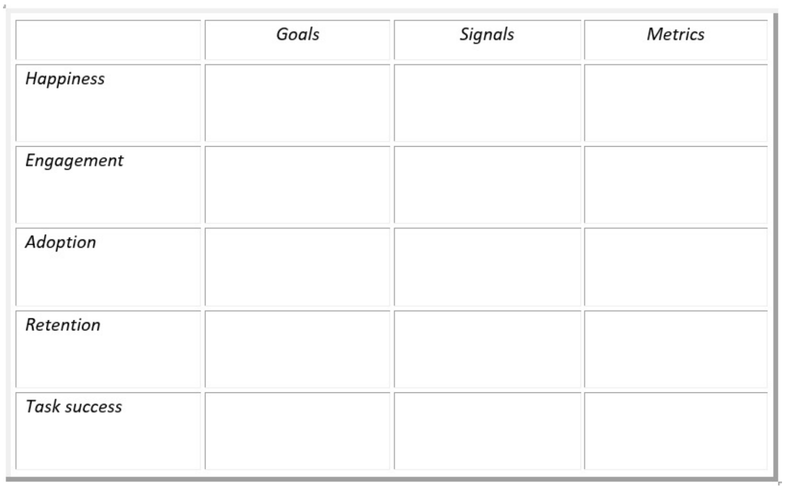
As one of the commonly used languages in the Web applications development, PHP has been chosen to be a scripting language for the study.

PHP script is meant to be developed performing three main tasks:

- collect patterns data based on the given input for starting element of the pattern, number of pattern steps and data slope
- process patterns and present analysed output to the output file
- perform statistical analysis on the given data, showing the result in the same output file

3.3.4 Measuring UX quality using HEART metrics

There is a large scope of Google developed HEART framework usages, as the metrics is flexible and easily customisable for the needs of the researcher. The best way to describe the general approach taken in the HEART framework towards producing informative outcomes for UX measurements, can be described in a well-known matrix or table, including metrics definitions and corresponding steps for their realization, see Figure 3.3.



	Goals	Signals	Metrics
Happiness			
Engagement			
Adoption			
Retention			
Task success			

Figure 3.3: HEART framework

According to the experience of the HEART developers mentioned in the article *How to choose the right UX metrics for your product* by Kerry Rodden [34], the approach of just picking a list of metrics for UX measurement and applying them to the given software does not produce a good result. Also, the metrics picking in this way would be highly subjective and will show the diversity of views towards this question inside the development team. Thus, the goals recognition is set as a first step.

The goals lay on higher level, on the surface, and recognize the primary thoughts what user-developer co-existence would define as an answer to the question "what this software is used for" or "what this application needs to do. Developer wise it would be the definition of success, e.g. our goal is to have users satisfied with easy to navigate and flexible to use interface. User wise it is something more subjective, but articulating the same issues, e.g. we want to be able easily and efficiently fill up the form and find the needed form on the first click.

It is obvious though that recognizing the goals of both developer and customer teams, and even coming to consensus on what the primary goal actually is, is not enough for getting the right flow of the development process, as the goal does not articulate the steps towards it neither gives the needed set of tools for the software builders team to achieve it. It clearly needs another level of abstraction, lower level of steps towards the goal. Here appears the "signals" level.

What user activity manifests for the development team? How their activity patterns show if the goal is met? Which steps they take would tell the developers that the goal user had is met? As Kerry Rodden [34] states, there always exist a large amount of potentially good signals or indicators for every defined goal. They can be a success signals or even a failure signal, both showing positive or negative statistics in walking towards the same goal. It is important to browse the list of indicators and get down to some number of most significant ones.

According to the same article [34], the good signal or indicator can be recognized by the following parameters. First of all, the signal must be easy to track, it should be easy to study logs for this signal, track it in large sets of data as efficient as in small amounts. Second, the focus on the dynamic development of the software must be recognized. The signal must be such that it would be sensitive enough to recognize, react and show something when the application design has been changed.

The final step to finalize the analysis framework is to define a set of metrics based on the signals defined before. Metrics for each indicator are also variable, as Kerry Rodden [34] states, and to be able to define some constant set, the data sample needs to be studied. After drawing parallels between data sample outcomes and signals set provided in the previous step of study, the good material for throughout analysis in a well defined stable set of metrics can be developed.

4 User Experience indicators in Raisoft application logs

Raisoft is a company that provides solutions for the interRAI Residence Assessment Instruments, used by heathcare sector and social services all over the world [31]. The solutions are used in elderly care, acute hare, child and youth care. The users are spread around the Globe. The application is highly modular, add-on modules can provide specific solutions for a variety of users and help in achieving big amount of versatile tasks, like completing assessment, defining care-plan, analysing data, managing resources, providing communication, helping with policy decisions. As interRAI is getting to be a standard in many health organisations, municipalities or even countries, Raisoft solutions are constantly evolving with the spread of the customer base.

4.1 User Experience realization in complicated web application

The current Raisoft application is complicated enough to produce the need for simplification for searching and performing the tasks needed for the user. The chance to log and follow the user activity patterns and analyse them using statistical and other methods can help the company to see how the software is actually used, find the ways to simplify the action sequences user has to take to perform some frequently made tasks, remove unusable modules and apply other possible enhancements to the application.

The discussion of UX enhancements possibilities has become more vivid and the tendency towards satisfying the customer experience with the software has appeared during the last few years, as the application received more (in amount and variety) customers.

On the other hand, development and SaaS teams have been returning to the question of using the existing logging system and a large amount of data collected by it in the way which would direct the development process towards enhancement of the software and user experience with it.

Recently, the idea to connect these two appeared to be a good problem - solution pair. For the UX experience needs and possibilities could be simply taken out of the logged user activity (or server calls activity) study.

In this study, the researcher would read the data from log files, filter by given input, which would include method of interest and number of steps following it, accumulate given sequences to learn the commonalities and find all possible user activity patterns in one big unit during considerably long period of time. The research would also collect some additional data as e.g. average response time, amount of sessions in month, length of the average session and such, to present better picture of the application usage.

The problems (or problem non-existence) with the UX in given application modules would then become obvious, and further actions can be taken to enhance the application module.

4.2 Test data sample

The decision on data used in the given study was based on the outcome of customer team questionnaire, see Appendix A, as well as the customer and development teams leaders suggestions. One of the largest customer was chosen to present the data scope more widely. The choice was based on the following reasons:

- the data reflects usage of the chosen array of interests defined by customer team
- the customer as an organization is experienced in the application use
- the amount of data is large enough to present bigger picture

The time scope of the data was set to one month, which after testing showed to be an optimal span, which accumulates all the usages of the application in a nutshell, as the application is usually worked with periodically (e.g. assessments can be filled and updated defined amount of times in a given period).

The data itself was chosen to be raw log files, which are built of the server calls occurring during the session. The reason behind it was an opportunity to learn whether the data presented in already existing log files is sufficient for mining the UX. Deliberately, no additional data or tools has been chosen to present optimal solution in simplicity. The approach would show the benefit of collecting this easily acquired data and using it for understanding the software capabilities and improvement needs.

The data presented in log files consists of both - server calls metrics and raw input/output messages. For the reasons of data security and non-disclosure, the files were filtered, and all possibly sensitive data was cleared out. The white-listed data included call or response metrics, process ids and a few call message pieces for several function calls, which were given as indicators for methods, that might be invoked in many different circumstances. For example, the call to get data from the server could be either complete case data or specific part of it.

The data is shared among two engines and filed daily. Consequent sessions are recorded in sequence into the same file. As such, the study went through 56 files, as the month of study was chosen to be February.

4.3 User Activity Learning Process

The process of realization of the UX has been done in two-sided manner, applying knowledge collected from the development team, software planners, architects and designers, and considering knowledge collected from the customer team and customer support team, who work with the end users hand-in-hand, as well as from the users themselves (not directly but through realisation of their activity). The collection of data in this step has been done in three steps.

First, questionnaire has been developed for the development and customer team that answered the following questions:

- what is general satisfaction level of the end users with the software (subjective opinion)
- what is the level of success of development team in realization of the user requests (subjective opinion)
- what are the main modules used by the end user, least at least 5
- what are the main features used by the end user, least at least 5
- what is the greatest strength of the current Raisoft web application
- what is the greatest weakness of the current Raisoft web application

This questionnaire produced needed results for understanding of the main modules and features to focus on in this study, helped to illustrate the gap between customer and development way of thinking, answered in a nutshell the question of the needs in improvements existing in current version of Raisoft Web application.

Secondly, the user activity data was collected from the logs. The data was limited to address the main issues listed in the questionnaire results and follow the progress of the modules and features mentioned in the questionnaire outcomes. This data has provided the needed base for studying the user activity patterns, which on their side, provided data for measuring the UX. The logs has been studied with help of such methods as statistical analysis and patterns discovery. First method, applied to greater scope of data, helped to limit it to the most used pieces of the software. Second method provided information for discovery of the ways how the modules and features are used, what are the routes of actions users take, are they comfortable with the features and modules or experience difficulties.

Third, the results collected from the logs and customer team replies has been compared with the ones defined by the development team. The development team has been asked for producing the set of estimations for each studied module and feature, including main task and its goal in the module or feature, defined set of steps the user should take to achieve the result, estimated time needed for succeeding, estimated frequency of usage of that feature or module for getting that task done.

After collection of the patterns data, it has been processed using PHP script. The decision on the use of PHP was based on the fact that the language is widely used in the company and is well known for the researchers. The script has been developed to being able processing large amounts of data.

The script reads given input file or files in a defined directory and collects sequences starting from the method of interest, defined in the input parameters. The sequences are of some defined length. The length is also provided as an input. After going through the files in question, the script calculates statistical data and processes the sequences into patterns, analyses the patterns and redirects results to the output files.

4.3.1 Data preparation

The data read from the log files needs to be preprocessed before the statistics is collected and patterns are formed and analysed. To prepare the data efficiently it is done along with the rest of functionality, meaning, during the process of reading the file line by line. To ensure the data being relevant, several filters are defined prior the script execution and then applied to each line of data.

One is the a filter of calls, the lines marked as server calls are the only one studied in this experiment.

Second filter is a filter for method relevance. Based on the given set of all the server functions, the set of functions or methods which are either purely server action related, metadata related or non-descriptive data collection calls are set to be ignored.

Yet another aspect of pattern collection filtering is filter by session. Only consequent steps within one session are to be considered a a sequence. The question of session boundaries is handled differently from software to software. As presented by K. Bovin [6], sessions are collections of interactions within the system limited by time and function. Thus, session becomes an analytical unit in the process of web software usage mining. As in log files, sessions are usually defined by either time gap (e.g. 30 minutes of inactivity), or session ids. In the given data, the id based session definition is used.

4.3.2 Data processing

After the filtering on the record is applied, a set of actions is performed on the selected record, which includes selection of the data for statistical and sequential patterns analyses.

In gathering of statistical data, the records present interest for calculating the following:

- total logged calls
- method of interest calls
- variable patterns amount
- average response time
- sessions in month
- average session time

The first two are simple counters, which get incremented ones the record of type CALL or a record of type CALL and of method id equal to the method of interest id is read. The average response time is calculated as mathematical mean of all the response time in milliseconds, written in RESPONSE records over the number of RESPONSEs.

Sessions are defined by session id. The opportunity to distinguish between sessions is only possible due to the existence of this id, since multiple concurrent sessions are often recorded at ones to the same log file. The moment a record with the new session id is met, it is recorded into the array of session with its unique id and timestamp marked as a starting time of the session. Every following record with the same session id is then marked as a temporary end record of the session with the timestamp as session temporary end time. Thus, ones the file is read, the very last occurrence of the record with the given session id will surely be the actual end of the session. Average session time is then calculated by first, finding the difference between session start and end timestamp, and then finding the mean of all the existing session times sum over number of unique session ids.

Sessions in months thus corresponds to the number of records in the session unique ids - session times array. Last,but not least, the variable patterns amount is calculated based on the data collected in the following step.

The following process is set to find and process patterns in the given file(s). As it is known, the pattern discovery can be either self-studied process or based on the proposition. In this research we focused on the self-study algorithm. The script receives an input value for sequence starting point and number of steps. Thus, the record are read according to these two parameters.

Firstly, if the record is a CALL and has the same method id as the input starting point method, the record is set as a first step of the new sequence, and recorded with a session id as an item key.

Then, each following item with the same session id as the start item is recorded to the same array as the following step of the sequence. This is done until the sequence steps are equal to the input number of steps. So, the check for sequence steps count is done on each recording. Ones the sequence is full, the if clause is executed, which sends the data of the sequence to processing and deletes it from the recorded sequences collection. The object reference for the processed patterns is sent along with it, as it gets filled gradually. The need for having recorded sequences cleared is the fact that large data sets will cause the memory drain otherwise. In all these steps we can rely on the records being ordered by the timestamp earlier first.

The function processing the sequences learns the patterns along the way. With each incoming sequence it compares it to the existing set and increments the counter on the existing pattern if already present, otherwise adds the new pattern setting its counter to 1. The output of the patterns collection is then redirected to the output file.

The result then has need studied by application of the UX metrics called HEART. The HEART chart has expressed the goals and metrics of the study and helped to show if the goals have been met according to the user activity analysis.

4.4 Scalability

The current Raisoft application is set to serve a great variety of user groups. As such, we can differentiate them by location (continent, country, region), work weight (number of cared persons per user) and job title (on which level given group of users are using the software). Based on the approach to define the groups of users, the scalability of data can be seen differently.

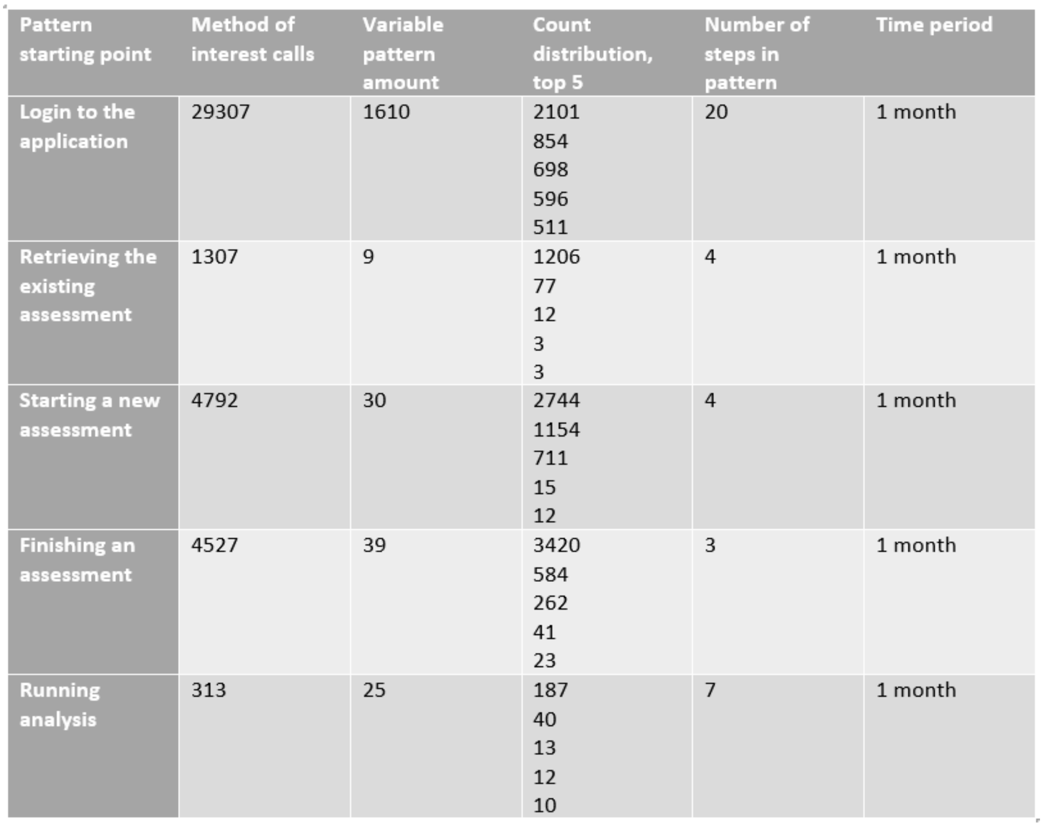
With the first type of user groups division, where we look at the data per county or location (with similar health care system, e.g. European countries), the data can be studied per location and the UX can be viewed and analyzed with the same scale for all the locations with similar health care system. With the recent course to globalization in all the existing systems including social, we can bravely assume that location-vise division in UX of the application usage will only decline in time. Thus, scalability of the research and applicability of statistical methods in this approach will be high.

Second approach of managing the user groups for analyzing the data, where we pay attention to the number of cared persons per unit or per user of the application, or simply, workload of the user considering the number of times he/she has to get back to the application and time spent using it, will not only help to analyze the current UX but also to assess the current quality of the software. As such, if the analysis will show that the same approach works well for all the users despite the number of persons they work with in relation to Raisoft software, we could consider the UX to be on a satisfactory level right away. This approach should also be applicable to the whole amount of data, if the current design is in satisfactory level and thus should be considered a highly scalable. Last and not trivial division of user groups, and thus the data related to their experience with the software, is the division based on the user role (or simply job description).

This approach splits the UAP data in well defined groups, where the way how the software is used, viewed and experienced is definitely different. In this approach, the scalability of the analysis of the data will be possible within the group. But with appearance of the new types of user groups the approach for study and analysis will have to be set differently.

5 Results and Analysis

Different sets of results have been acquired during the user activity research and patterns discovery from the server logs. The results depend on the number of sequence steps, time period, method of interest. The diversity of the results has been summarised in the Table 5.1. As it can be seen, each method has its own presentation in the activity chart of the application, thus has to be also analysed and studied separately.



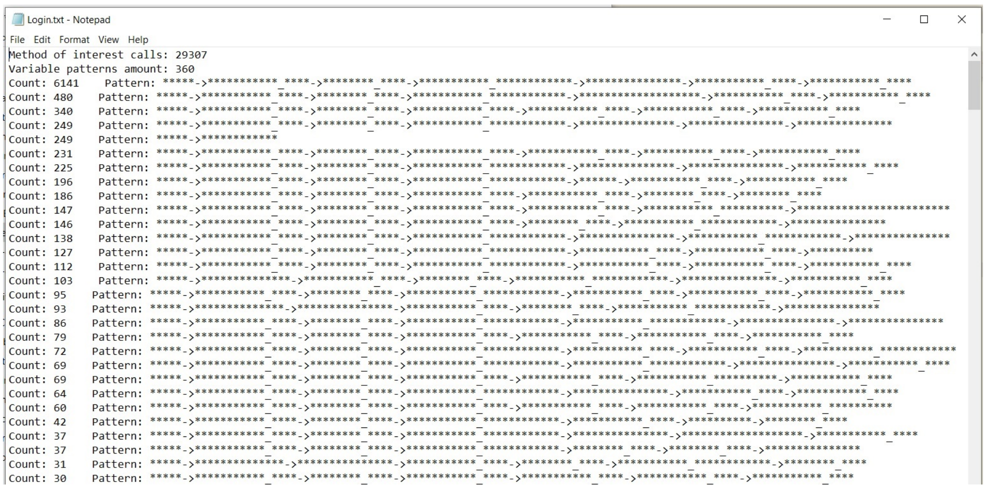
Pattern starting point	Method of interest calls	Variable pattern amount	Count distribution, top 5	Number of steps in pattern	Time period
Login to the application	29307	1610	2101 854 698 596 511	20	1 month
Retrieving the existing assessment	1307	9	1206 77 12 3 3	4	1 month
Starting a new assessment	4792	30	2744 1154 711 15 12	4	1 month
Finishing an assessment	4527	39	3420 584 262 41 23	3	1 month
Running analysis	313	25	187 40 13 12 10	7	1 month

Table 5.1: Patterns discovery results example

5.1 Output

The output of the trials were recorded into the text files, one for each method under study. As the study units were Login module and Assessment module, as well as the known by the customer support team as a problematic one, Analysis module also known as Quality module. A separate file with statistical data was produced as well.

The files on modules included two measurements: method of interest calls and variable patterns amount, as well as a list of patterns, consisting of each pattern occurrence count and its structure. The output form was as following 5.1:



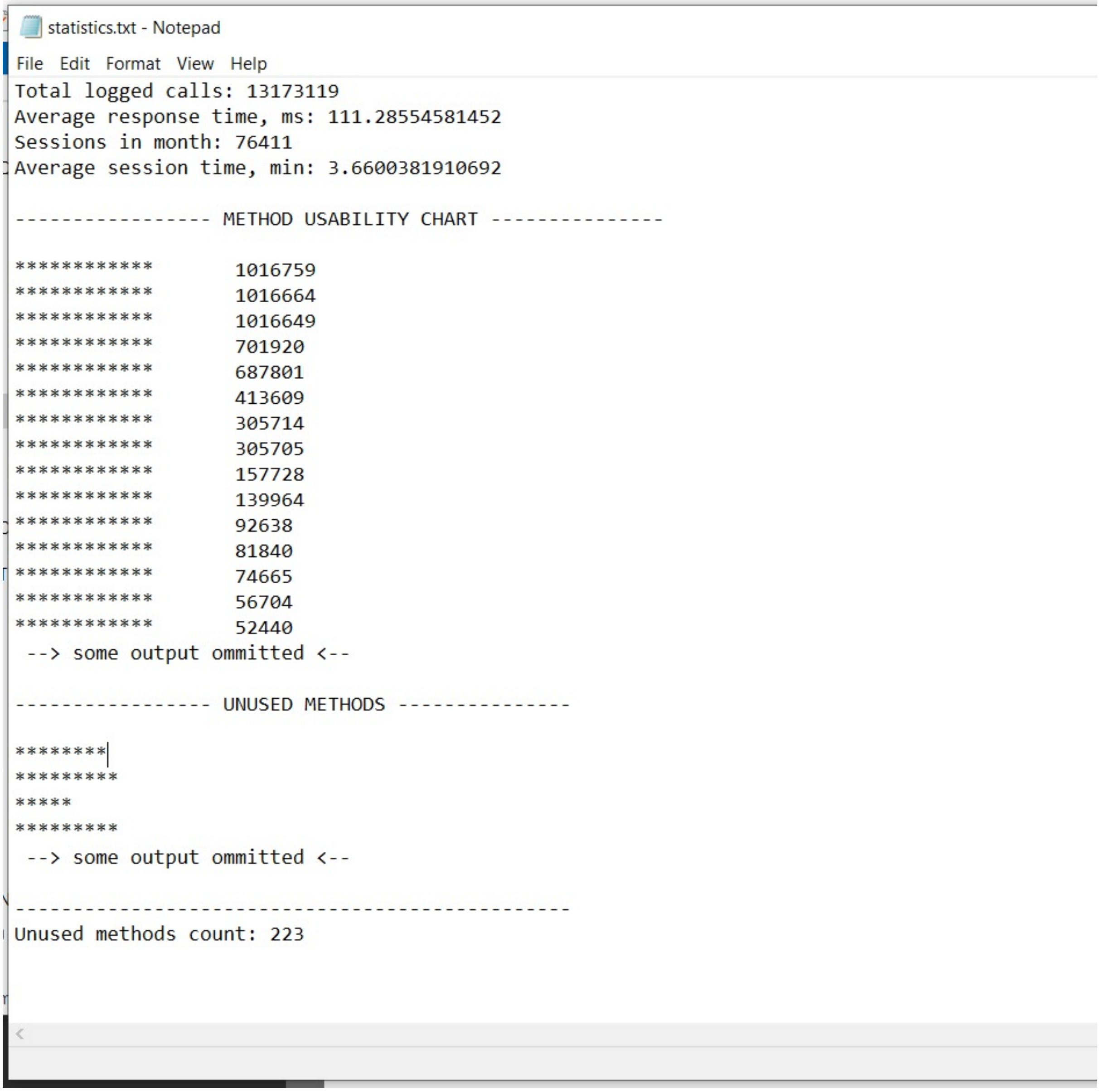
```
Method of interest calls: 29307
Variable patterns amount: 360
Count: 6141    Pattern: *****->***********_****->*********_****->***********_***********_************->***************->***********_****->***********_****
Count: 480    Pattern: *****->***********_****->*********_****->***********_***********_************->******************->***********_****->***********_****
Count: 340    Pattern: *****->***********_****->*********_****->***********_****->***********_****->***********_****->***********_****->***********_****
Count: 249    Pattern: *****->***********_****->*********_****->***********_***********_************->***************->***************->***************
Count: 249    Pattern: *****->************
Count: 231    Pattern: *****->***********_****->*********_****->***********_****->***********_****->***********_****->***********_****->***********_****
Count: 225    Pattern: *****->***********_****->*********_****->***********_***********_************->***************->***************->***********_****
Count: 196    Pattern: *****->***********_****->*********_****->***********_***********_************->******->***********_****->***********_****
Count: 186    Pattern: *****->***********_****->*********_****->***********_****->***********_****->***********_****->*********_****->*********_****
Count: 147    Pattern: *****->***********_****->*********_****->***********_****->***********_****->***********_****->***********_**********->***********************
Count: 146    Pattern: *****->***********_****->*********_****->***********_****->***********_****->*********_****->***********_************->***************
Count: 138    Pattern: *****->***********_****->*********_****->***********_***********_************->***************->***********_************->**************
Count: 127    Pattern: *****->***********_****->*********_****->***********_***********_************->***********_****->***********_****->**********
Count: 112    Pattern: *****->***********_****->*********_****->***********_***********_************->***********_****->***********_****->***********_****
Count: 103    Pattern: *****->**************->***********_****->*********_****->***********_****->***********_************->***************->***********_****
Count: 95    Pattern: *****->***********_****->*********_****->***********_***********_************->***********_****->***********_****->***********_****
Count: 93    Pattern: *****->**************->***************->***********_****->***********_****->*********_****->***********_************->***************
Count: 86    Pattern: *****->***********_****->*********_****->***********_***********_************->***********_************->***************->***************
Count: 79    Pattern: *****->***********_****->*********_****->***********_****->***********_****->***********_****->***********_****->***********_****
Count: 72    Pattern: *****->***********_****->*********_****->***********_***********_************->***********_****->***********_****->***********_************
Count: 69    Pattern: *****->***********_****->*********_****->***********_***********_************->***********_************->***************->***********_****
Count: 69    Pattern: *****->***********_****->*********_****->***********_****->***********_****->***********_****->***********_**********->***********_****
Count: 64    Pattern: *****->***********_****->*********_****->***********_***********_************->***************->***********_****->***********_****
Count: 60    Pattern: *****->***********_****->*********_****->***********_****->***********_****->***********_****->***********_****->***********_**********
Count: 42    Pattern: *****->***********_****->*********_****->***********_***********_************->***********_****->**********->*********_****
Count: 37    Pattern: *****->***********_****->*********_****->***********_***********_************->***************->*****************->***********_****
Count: 37    Pattern: *****->***********_****->*********_****->***********_***********_************->*********_****->*********_****->***************
Count: 31    Pattern: *****->**************->***************->***********_****->***********_****->*********_****->***********_************->*********_****
Count: 30    Pattern: *****->***********_****->*********_****->***********_****->***********_****->***********_****->***********_****->***********_****
```

Figure 5.1: Pattern output file structure

General statistics was recorded into separate file. The total number of calls during the month, average response time, number of sessions in month and average session time are presented in the statistics file. The data also included usability chart for all the methods presented in the application of the following structure 5.2:

5.1.1 Login patterns

Login is one of the most appearing server calls, since the user is only allowed to proceed with the application after authorisation. There were 29307 login calls in the month of February 2021 (from now on all the calculations are presented for the specific user unit, a large healthcare organisation). Comparing to the total amount of logged server calls in this month equal to 13173119, it is 2 percent of the total.



```
Total logged calls: 13173119
Average response time, ms: 111.28554581452
Sessions in month: 76411
Average session time, min: 3.6600381910692

----------------- METHOD USABILITY CHART ---------------

************       1016759
************       1016664
************       1016649
************       701920
************       687801
************       413609
************       305714
************       305705
************       157728
************       139964
************       92638
************       81840
************       74665
************       56704
************       52440
 --> some output ommitted <--

----------------- UNUSED METHODS ---------------

********
*********
*****
*********
 --> some output ommitted <--

------------------------------------------------
Unused methods count: 223
```

Figure 5.2: Statistics output file structure

This is a high value, considering the fact, that the total amount of variable logged methods at the moment of study is 385.

In order to study the Login patterns, the amount of after login steps has been changed to see the optimal outcome, which would indicate the tendencies of the user activities in the first few minutes after authorisation. As it showed to the testers, the optimal number of server calls after login was 7. The reason why having only 2 or 3 server calls after login would not show the actual direction of the user activity is the fact that one user action (for example click or navigation), has several corresponding server calls happening in so called clusters. An illustration for that would be click on login button, which would then make server calls related to authorisation checking and perform the actual login into the system. Also, the user might login in point of session timed out. After that login click, the application will automatically open the page, the user has been signed out from, thus, another cluster of corresponding calls will appear without actual user interaction. Thus, 7 after Login steps were studied. Also, to produce more precise numbers, 2 and 20 after login steps were also studied for comparison.

As amount of patterns is directly proportional to the number of steps setup, as can be seen from the Table 5.1 in the pattern search input, the number of patterns with 7 consequent steps happen to be much larger (360 patterns) than the number of those with 2 items (10 patterns).

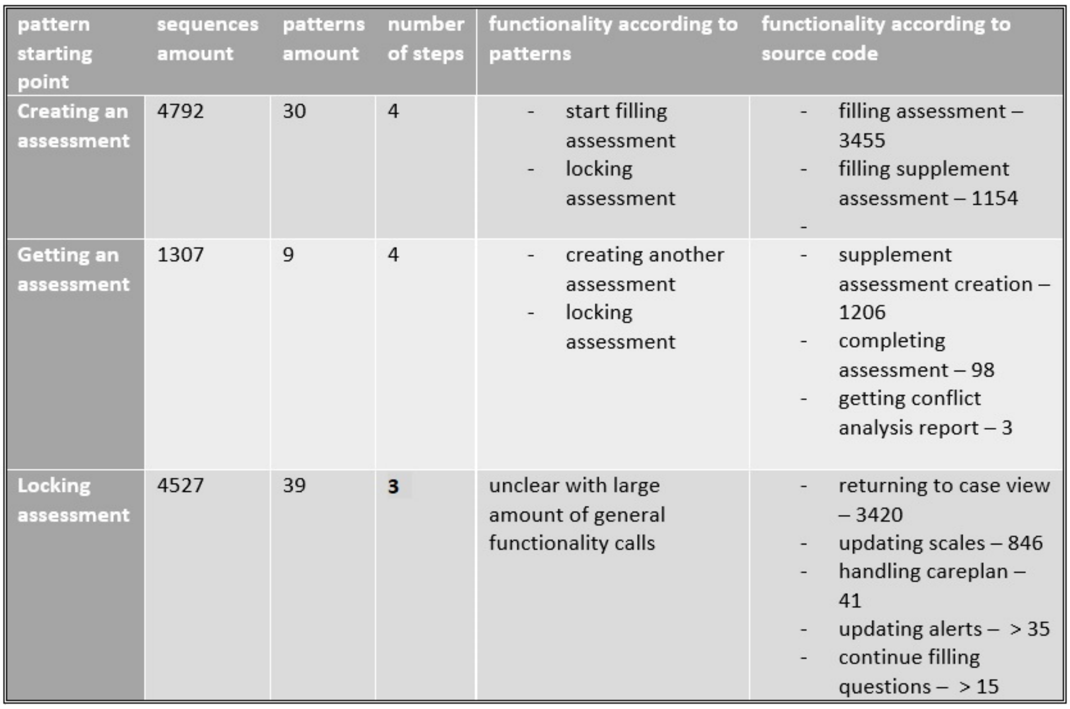
The amount of variable patterns of 7 sequential calls starting from Login in the month of February 2021 was 360. The most used pattern appeared 6141 times and included navigation to the case (with the following step of retrieving an assessment as the further study showed). The following pattern by number of occurrences (480) showed the same tendency of the user to retrieve the needed case to proceed with it, but in this pattern instead of navigating to the specific case from the ward case list, the navigation was done using the case search. The following items at most repeated the items of these two patterns with either different order or in variable amount, e.g. performing several ward case list retrieving operations or switching between cases variable number of times.

Some of the patterns though could not be analysed in this schema as they have invoked the functionality which has been deprecated. The method in question is generic and specification of the call is only described as an input parameter, which for the case of this study has been blacklisted as the one which might hold some sensitive data. The amount of that calls were 12531.

440 logins are followed by a password change, 462 resulted in authentication check (wrong login), 153 times contact info is retrieved right after login and in 19 cases, the user changes language.

5.1.2 Assessment patterns

Assessment module study was performed by reading the sequences following assessment creation, call to get an existing assessment and also which appear after the assessment has been completed (locked). These are three common activities done in the assessment module.



pattern starting point	sequences amount	patterns amount	number of steps	functionality according to patterns	functionality according to source code
Creating an assessment	4792	30	4	- start filling assessment - locking assessment	- filling assessment – 3455 - filling supplement assessment – 1154 -
Getting an assessment	1307	9	4	- creating another assessment - locking assessment	- supplement assessment creation – 1206 - completing assessment – 98 - getting conflict analysis report – 3
Locking assessment	4527	39	**3**	unclear with large amount of general functionality calls	- returning to case view – 3420 - updating scales – 846 - handling careplan – 41 - updating alerts – > 35 - continue filling questions – > 15

Table 5.2: Assessment related patterns results

The outcome of the assessment module study became questionable, due to the fact that the data linked to these server calls has been completely anonymised. For example, it is impossible to see which assessments has been created or completed within some sequence, as the assessment id was not visible for the course of this study. The questionable outcomes were researched deeper with the aid of source codes, to give an idea, which patterns correspond to specific user activities on the module.

As such, the outcome can be seen from the Table 5.2. As it shows, the behaviour of the user and the pattern of his/her actions are not clearly visible, especially after the assessment has been locked, as there are large number of calls to generic functions on server that can only be defined based on the input and output data. With help of analysis of the source code, the better view on the user actions was obtained. As the table describes, assessments has been created and locked more than 4500 times during the month, and accessed little over 1300 times during the same period of time.

5.1.3 Analysis patterns

Analysis is the module marked by the technical support team as the most complicated one for the user. "Quality module has been one of the most problematic areas of the application. There has been random bugs here and there varying from being unable to save analysis to some more minor ones like typos. User problems are usually hardships to understand the ruling system since it can seem quite complex at first since it has many variables." - customer support team.

To see if the problems with the module have some underlying module design or implementation issues, the logs has been studied for any abnormal activity of the user after accessing analysis.

The pattern were formed as 7 steps starting from the getting analysis call. The module is designed so, that the user gets access to the list of existing analysis with the possibility to either view existing or creating new one. Analysis list has tagging option, which allows to add either predefined or user specified tags to the items in order to navigate within the list easier.

As the study shows, the analysis has been retrieved 313 times during the February of 2021. The user activity then took quite expected and non-highly variable pattern. The amount of variable patterns was 7, with majority (187) of getting the analysis report and adjusting some values within it to presumably run it again. 71 times within 313 the tagging system was used.

5.2 Elaboration on the collected data

The study of patterns relied highly on the sequence length, thus variable lengths have been applied to each case, the outcome summary can be seen from the figure

5.3. The direct proportion of the amount of sequence steps and amount of the patterns is obvious, as well as the dependency of the patterns amount and feature in question. For example, the actions after assessment creation are more predictable than the ones after the assessment closing. Because opening the assessment has a clear goal of filling or changing the record, when closing it may have any goals from continuing with the work flow to quitting the daily routine. That is why after Login and after Assessment closing patterns grow highly with the sequence length growth, while patterns following the Analysis and Assessment opening are more predictable as have more defined goals and actions flow.

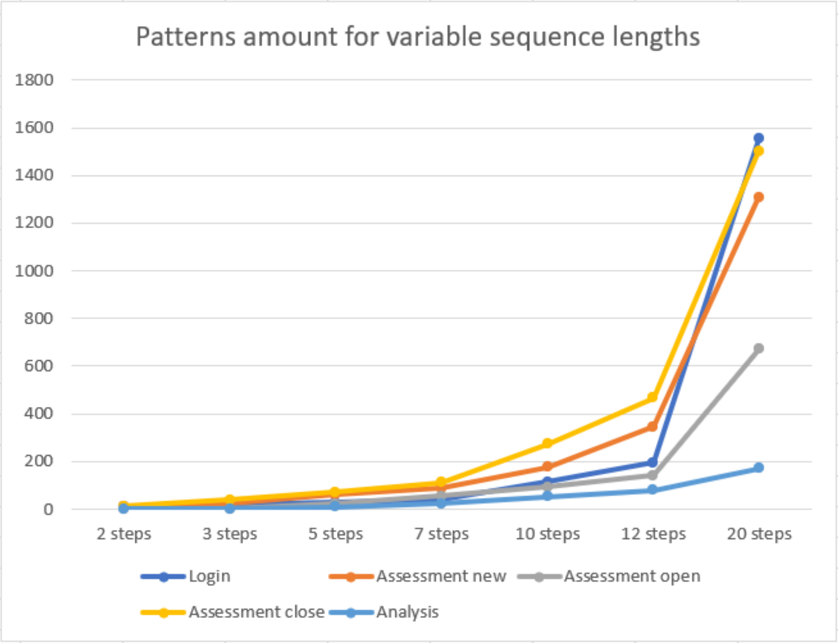


Figure 5.3: Patterns amount for variable sequence length

5.2.1 Login patterns analysis

After Login patterns distribution can be seen from the following Figure 5.4. The chart displays the obvious prevalence of one or two main patterns over the variety of the secondary ones, despite the length of the sequences.

General picture of the Login patterns displays obvious activity of the user navigating to the case and assessment right after login to perform needed tasks. As it has been designed, on login, the user needs to get a view with the organisation structure and the search possibilities, transformed now into new dashboard view, which

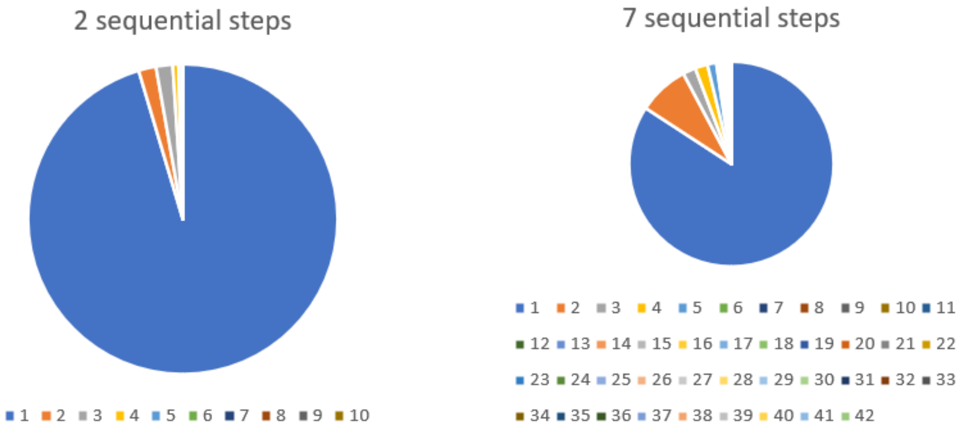


Figure 5.4: After Login Patterns distribution

provides user with more and customisable options on the initial view. A few facts, which might raise a question on UX drawbacks are password change and wrong login credentials frequency.

5.2.2 Assessment patterns analysis

The majority of assessment creation occurrences were followed by assessment filling or assessment locking, while accessing the existing assessment continued by locking an assessment or creating another one. The behaviour of locking right after creation is questionable as it was set to be filled up first. After some investigation, the conclusion was driven to the idea, that another assessments (support assessments within the original lengthy one) are created during the filling of the original one, creating the situation where creation and retrieving of the assessment is followed closely by the lock.

As well, patterns after lock are always end up in large number, which can be explained by firstly, a variety of assessments filled, which update different items in the case, and secondly, by the fact, that often assessment lock happens on the concurrently filled smaller support assessment, resulting in another assessment filling continuation right after it.

As the study showed, reading and analysing assessment related behaviour of the user and system is not easy, and with the limited input data, it is nearly impossible. The research was decided to be stopped on general outcome, as speculation on the specific behaviours would possibly create false estimations and conclusion.

5.2.3 Analysis patterns analysis

The distribution of the analysis patterns with 5 and 7 sequential steps is presented in the Figure 5.5. The 2 and 3 sequential steps resulted in only 1 pattern, as in some cases, the amount of server call steps does not correspond directly to the user actions, and for some modules one user click results in several server call steps. Again, a few main patterns prevail the variety of the secondary ones, which indicates the existence of the well defined action flow within the module in question.

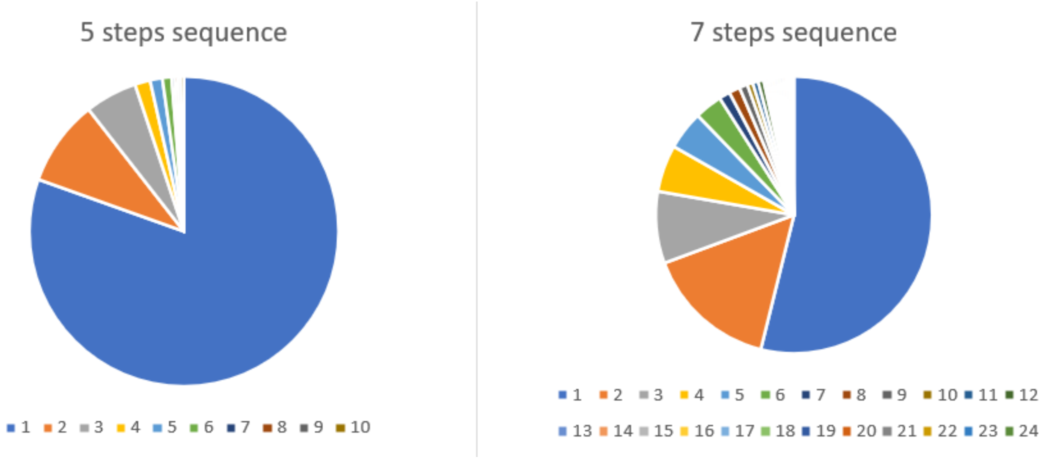


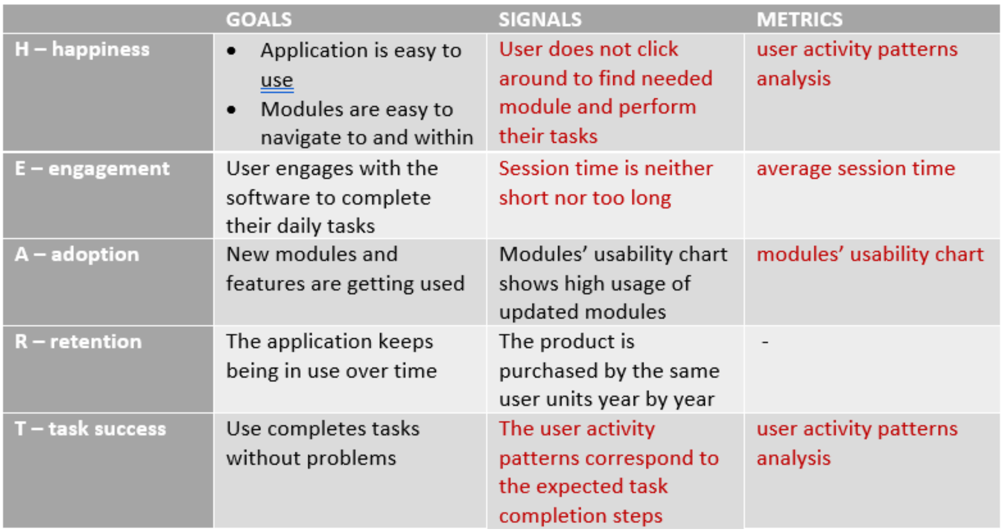
Figure 5.5: Analysis module Patterns distribution

Analysis patterns have not revealed the problems, mentioned by the users to the support team. It has become obvious that the possible issue with the module is created by it's versatility and complexity rather than by the inability to perform the task efficiently. It also helped to understand that not all the problems in UX are laying in the area of functionality, accessibility or efficiency of the application, but can be a subject of general feeling with the unit usage.

5.2.4 Measuring the HEART rate

The HEART metrics has been applied to access the UX quality of the software in question using UAP analysis and basic statistical analysis of the log files data. The following table 5.3 presents the metrics tool:

As it can be seen from the Table 5.3, highlighted items (H, E and T) has been studied and presented in the given research. The only issue, which cannot in full be realised with help of User Activity Patterns discovery and analysis of the log files,



	GOALS	SIGNALS	METRICS
H – happiness	• Application is easy to use • Modules are easy to navigate to and within	User does not click around to find needed module and perform their tasks	user activity patterns analysis
E – engagement	User engages with the software to complete their daily tasks	Session time is neither short nor too long	average session time
A – adoption	New modules and features are getting used	Modules' usability chart shows high usage of updated modules	modules' usability chart
R – retention	The application keeps being in use over time	The product is purchased by the same user units year by year	-
T – task success	Use completes tasks without problems	The user activity patterns correspond to the expected task completion steps	user activity patterns analysis

Table 5.3: Applying HEART metrics

was the retention, as the application is not a matter of choice for the end user, but rather the decision of the organisation or residence the user is working in.

As for happiness and task success the UAP analysis are the metrics used in assessment. And, according to the outcomes of the given study, the modules in question have been used according to the expectations in majority of the cases and the tasks performed within the modules have been obvious. There was no significant variations of the patterns, as the biggest difference was often presented in the steps order rather than in the items themselves. After Login patterns showed to be the ones navigating to case or assessment, after Assessment patterns resulted in assessment filling or locking, and Analysis module patterns presented user activity in navigation within the analysis list using tagging system and running the analysis as they are or with slight adjustments. As it can be seen from the patterns analysis, the user in general navigated to and within the modules in obvious manner and performed the tasks with no unusual steps within the sequences. Ergo we can assume, the first and the least items in HEART metrics can be marked as success.

Engagement of the user while using the software to solve tasks is best to be seen in the time the user spends with the application in question. As for the subject of this study, the time elapsed in one session is the measure for that. According to the system requirements, user session will expire if there is no activity for more than 30 minutes. Otherwise, the session will basically run from login and until logout. After statistical analysis of the given logs' data, the average session time is equal to 13.67 minutes, varying from just a few minutes to 76.42 or even 162.6 minutes sessions. Such difference can be easily explained by the variety of the tasks, which can be performed with help of the application in question. Some, as printing pdf report, can take just a minute, while other, as filling an assessment, can take more than an hour. However, the average session time is a good indicator on the user involvement with the application, and it also indicates that the software is fairly complicated. Based on the developers team opinion, overcomplicated application would involve the user for more than 10 minutes sessions. Thus, the Engagement metrics of the UX produce positive result.

Adoption signals are not highlighted in the Table 5.3, as they cannot be fully tested with the logs provided by just one user group. The software in question is versatile and serve the needs of variable customers all over the world. Thus, single customer uses only part of the available tools, which then affects the logs to include request calls to only section of available server functions. So, partially, it is possible to study the adoption of the new ideas and changes by clearly defining which modules and functionality usage we are looking for. For example, the developers can pay attention to the fact that password change occurred 458 times during the month, while username change - only 12, this functionality is general for all the users, so, it can be assessed based on the one unit logs.

Concluding the results of applying HEART metrics onto the UAP and log based statistics, we can state, that the UX level of application functionality and modules' structure is satisfying, as 3 goals of the software development has been achieved, according to the UAP analysis and logs data statistics.

6 Discussion

The performed study has been enlightening experience, as it both - showed possibilities for exploring and enhancing User Experience using existing sets of tools available on site of the development process, and showed the issues with lack of development - user understanding and co-existing in the design and development process, as well as the needs for better logging structure for clearer analyses outcomes.

The work can be considered successful in terms of engaging the UAP and the UX within one process of learning and analysing the application, and proving the concept of possibility to learn significant amount of user experience related issues from the existing log files with the existing tools of scripting, statistics, metrics. A plan of learning the usage of several modules based on existing logging has resulted in generally expected outcomes, giving a few red lights along the way, showing the need to improve logging, verbalise the process.

The drawbacks of the study lied mostly in the area of the log files structure and anonymisation needs. Due to the limited resources, the study could not produce more precise results for the company use, but rather served as a prove of concept for existing possibility to involve the UAP analysis in the UX enhancement process.

The biggest challenge of the work was to apply the pattern search and analysis script to the existing data, as the row log data could not produce reliable results in its original form. For example, when learning a sessions structure and a life cycle, it required additional learning and adjusting of the data scope. As well, amount of server calls unrelated to the user activity had to be recognised and set to be ignored. Data, which has been removed for the security purposes, has affected the outcome of the study as well. Specifically requested items for the study, such as usability of some specific features, like search or printing, could not have been studied at all, as some of the features are not logged properly. That taught us about the need to pay more attention to the logging system of the application, provide more descriptive records without disclosing sensitive data, make sure to report all the activity to the server with the existing logging methods.

However, the research has proven the usability of the User Activity Pattern analysis from the log files and collection of the statistics from the logging for the estimations of the success of the application design as well as the source of information on the UX positive and negative tendencies. Using the given data, the study has proven that the application in question serves its needs in at least 3 modules, is generally well used and, even though has a room for improvement, still meets the needs of the end user.

HEART framework has proven to be a comfortable UX measurement tool, which provided the research with the needed set of goals, signals and metrics for the User Experience assessment. The framework could be a good addition to the design and development process, and serve as a discussion plate for the development - customer team communication.

The performed analysis are scalable, meaning, can be performed on a different data amount or for the variable time gap. Similar study can be applied also to the different users group, but it will require different module choices, based on the user group, user roles, installed modules and available features. The analysis are not location-specific, data-size-specific or unit-type-specific. As such, they can possibly be automated and run periodically, analysing activity in different modules, adjusted by some input parameters. As well, a future study can focus on the algorithms, which can be applied to the given UAP data for behaviour predictions and anomaly detections. Behaviour predictions could help to customise the software for specific user groups, while an anomaly detection can help to see the issues with the design or development at the first stages of its implementation for general use.

7 Conclusion

User Experience has become a contemporary point of interest in a web application development. Unlike old approaches to the evaluation of the product success, UX answers the questions, which are completely different in nature comparing to the ones, the technological world is used to. As such, common software quality assessment instruments cannot be applied to the measurement of the User Experience. This research presented User Activity Patterns analysis based on the server logs as one of the simple yet powerful solutions for the UX measurement. The positive aspects of this approach include simplicity, availability and the fact that the user can be involved in the process of development indirectly.

HEART framework has proven to be a good help tool, providing the researcher with a defined set of goals, signals and metrics for the UX assessment. The tool is proposed for the later use in the design or development process as well in discussions between the teams within the company as well as between developers and users.

The idea of focusing on the user within the development process has been found interesting and applicable to the web development process. It is generally true, that "developing *for* people" must go hand-in-hand with "developing *with* people", and it does not always require direct human interaction, as this research proved. Paying attention to the user activity and studying his/her involvement with the software, adoption of the new features, retention and engagement with the product as well as measuring the task completion success using logging systems available, can create a strong knowledge-base for the developers and designers to build their ideas for the development process on.

References

[1] ALLAM, A., AND DAHLAN, H. M. User experience: Challenges and opportunities. *Journal of Information Systems Research and Innovation 3*, 1 (2013), 28–36.

[2] ASAD, N. Human-centered design: 9 examples to justify why it matters, 2019. URL `https://www.wowmakers.com/blog/9-human-centered-design-examples/`, accessed 17.05.2021.

[3] BANDURA, A. Self-efficacy. *The Corsini encyclopedia of psychology* (2010), 1–3.

[4] BARAGLIA, R., AND SILVESTRI, F. Dynamic personalization of web sites without user intervention. *Communications of the ACM 50*, 2 (2007), 63–67.

[5] BATTARBEE, K., ET AL. *Co-experience: understanding user experiences in interaction*. Aalto University, Helsinki, Finland, 2004.

[6] BØVING, K. B., AND SIMONSEN, J. Http log analysis: an approach to studying the use of web-based information systems. *Scandinavian Journal of Information Systems 16*, 1 (2004), 3.

[7] CARPENTIER, T., AND LAMBERT, T. *Beautiful users: Designing for people*. Chronicle Books, San Francisco, USA, 2014.

[8] CHEKANOV, S. V. *Scientific data analysis using Jython Scripting and Java*. Springer Science & Business Media, Berlin/Heidelberg, Germany, 2010.

[9] COLE, M. J., HENDAHEWA, C., BELKIN, N. J., AND SHAH, C. User activity patterns during information search. *ACM Transactions on Information Systems (TOIS) 33*, 1 (2015), 1.

[10] CONRATH, D. W., AND MIGNEN, O. P. What is being done to measure user satisfaction with edp/mis. *Information & Management 19*, 1 (1990), 7–19.

[11] COSTABILE, M. F., FOGLI, D., MUSSIO, P., AND PICCINNO, A. End-user development: The software shaping workshop approach. In *End user development*. Springer, 2006, pp. 183–205.

[12] DAVIS, F. D. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly 13*, 3 (1989), 319–340.

[13] DESMET, P. M., AND HEKKERT, P. Framework of product experience. *International journal of design 1*, 1 (2007), 57–66.

[14] FLAOUNAS, I., AND KOKKINAKI, A. On the evaluation of oles using the heart framework. *Ascilite* (2015), 668–671.

[15] FOX, D., SILLITO, J., AND MAURER, F. Agile methods and user-centered design: How these two methodologies are being successfully integrated in industry. In *Agile 2008 Conference* (2008), IEEE, pp. 63–72.

[16] GRUDIN, J. Managerial use and emerging norms: effects of activity patterns on software design and deployment. In *37th Annual Hawaii International Conference on System Sciences, 2004. Proceedings of the* (Jan 2004), IEEE, p. 10.

[17] HASSENZAHL, M., AND TRACTINSKY, N. User experience-a research agenda. *Behaviour & information technology 25*, 2 (2006), 91–97.

[18] HEINBOKEL, T., SONNENTAG, S., FRESE, M., STOLTE, W., AND BRODBECK, F. C. Don't underestimate the problems of user centredness in software development projectsthere are many! *Behaviour & information technology 15*, 4 (1996), 226–236.

[19] HUSSAIN, J., KHAN, W. A., HUR, T., BILAL, H. S. M., BANG, J., HASSAN, A. U., AFZAL, M., AND LEE, S. A multimodal deep log-based user experience (ux) platform for ux evaluation. *Sensors 18*, 5 (2018), 1622.

[20] JOSHI, M., KARYPIS, G., AND KUMAR, V. A universal formulation of sequential patterns. URL `https://www.researchgate.net/publication/2509514_A_Universal_Formulation_of_Sequential_Patterns`, accessed 02.02.2020.

[21] KAZMAN, R., AND BASS, L. *Categorizing business goals for software architectures*. Tech. Rep. CMU/SEI-2005-TR-021, Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst, 2005.

[22] LAW, E. L.-C., ROTO, V., HASSENZAHL, M., VERMEEREN, A. P., AND KORT, J. Understanding, scoping and defining user experience: A survey approach.

In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2009), CHI ?09, Association for Computing Machinery, pp. 719–728.

[23] LUND, A., AND LUND, M. Descriptive and inferential statistics, 2018. URL `https://statistics.laerd.com/statistical-guides/descriptive-inferential-statistics.php`, accessed 12.02.2021.

[24] MENGE, L. Articulate agile, 2019. URL `https://articulateagile.com/experience/apple-inc`, accessed 17.05.2021.

[25] MOGGRIDGE, B., AND ATKINSON, B. *Designing interactions*, vol. 17. MIT press Cambridge, MA, London, England, 2007.

[26] MORZY, T., WOJCIECHOWSKI, M., AND ZAKRZEWICZ, M. Fast discovery of sequential patterns using materialized data mining views. In *Proceedings of the 15th ISCIS Conference* (Poznan, Poland, 2000).

[27] NIRKHI, S., AND HANDE, K. A survey on clustering algorithms for web applications. In *Proceedings of the 2008 International Conference on Semantic Web and Web Services, SWWS 2008*, pp. 124–129.

[28] NORMAN, D., AND NIELSEN, J. Iso 9241-210:2019 ergonomics of human-system interaction - part 210: Human-centred design for interactive systems, 2019. URL `https://www.iso.org/obp/ui/#iso:std:iso:9241:-210:ed-1:v1:en`, accessed 17.05.2021.

[29] NORMAN, D., AND NIELSEN, J. The definition of user experience (ux), 2021. URL `https://www.nngroup.com/articles/definition-user-experience/`, accessed 14.05.2021.

[30] PEARL, J. Bayesian networks, 2011. URL `https://escholarship.org/uc/item/53n4f34m`, accessed 24.05.2021.

[31] RAISOFT. Raisoft ltd, 2021. URL `https://www.raisoft.com/en/frontpage/`, accessed 17.05.2021.

[32] RAJAK, A., AND GUPTA, M. K. Association rule mining: applications in various areas. In *Proceedings of international conference on data management, Ghaziabad, India* (2008), pp. 3–7.

[33] RICE, J. A. *Mathematical statistics and data analysis*. Thomson Brooks/Cole, Belmont, CA USA, 2007.

[34] RODDEN, K. How to choose the right ux metrics for your product, 2015. URL `https://library.gv.com/how-to-choose-the-right-ux-metrics-for-your-product-5f46359ab5be`, accessed 25.02.2020.

[35] RODDEN, K., HUTCHINSON, H., AND FU, X. Measuring the user experience on a large scale: user-centered metrics for web applications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2010), ACM, pp. 2395–2398.

[36] SHARMA, A. K., AND SAHNI, S. A comparative study of classification algorithms for spam email data analysis. *International Journal on Computer Science and Engineering 3*, 5 (2011), 1890–1895.

[37] SRIVASTAVA, J., COOLEY, R., DESHPANDE, M., AND TAN, P.-N. Web usage mining: Discovery and applications of usage patterns from web data. *Acm Sigkdd Explorations Newsletter 1*, 2 (2000), 12–23.

[38] THOMAS, S., AND SARAWAGI, S. Mining generalized association rules and sequential patterns using sql queries. In *KDD Knowledge Discovery and Data Mining (International Conference)* (1998), AAAI, pp. 344–348.

[39] TONELLA, P., AND RICCA, F. Dynamic model extraction and statistical analysis of web applications. In *Proceedings. Fourth International Workshop on Web Site Evolution* (2002), IEEE, pp. 43–52.

[40] WAGNER, E. L., AND PICCOLI, G. Moving beyond user participation to achieve successful is design. *Communications of the ACM 50*, 12 (2007), 51–55.

[41] WOJCIECHOWSKI, M. Mining various patterns in sequential data in an sql-like manner. In *ADBIS '99: Proceedings of the Third East European Conference on Advances in Databases and Information Systems (Short Papers)* (September 1999), Citeseer, pp. 131–138.

[42] WU, C., XU, B., SHI, S., AND ZHAO, B. Time–activity pattern observatory from mobile web logs. *International Journal of Embedded Systems 7*, 1 (2015), 71–78.

A Customer team questionnaire answers

- Please answer the following questions. 1 = very bad 5 = very good
  - What is general satisfaction level of the end users with the software (subjective opinion)? /Subjektiivinen näkemyksesi: Kuinka tyytyväisiä ohjelmiston loppukäyttäjät yleisesti ottaen ovat ohjelmistoon?
    - 4 4 4 4 4 3 5 **overall: 4**
  - What is the level of success of development team in realization of the user requests (subjective opinion)? Subjektiivinen näkemyksesi: Kuinka hyvin kehitystiimi ymmärtää/oivaltaa/tajuaa asiakkaiden pyynnöt?
    - 3 3 4 3 3 3 2 **overall: 3**
- What are the main modules used by the end user, least at least 5? / Mitä allaolevista sivuista/näkymistä loppukäyttäjät käyttävät eniten. Valitse vähintään 5.

assessment / arviointinäkymä	7
case reports / henkilöraportit	6
login / kirjautumisnäkymä	6
main / päänäkymä	6
manage users / käyttäjien hallinta	3
person / henkilönäkymä	3
compare assessment / vertaile arviointeja	2
report next complete assessment / seuraavat kokonaisarvioinnit	2
manage organisation / organisaation hallinta	1
report helsa / asiakastason tulokset -raportti	1
report organization statistics / organisaation tilastot	1
report unlocked assessments / llukitsemattomat arvioinnit	1
case background info / henkilön taustatiedot	1
analysis tools (aka quality module) / laatumoduuli	1
case / henkilön jaksonäkymä	1
dashboard / työpöytä	1
manage roles / roolien hallinta	1

- What are the main features used by the end user, least at least 5? / Valistunut arvauksesi: Mitä ao. ominaisuuksista arvelet ohjelmiston käyttäjien käyttävän eniten? Valitse vähintään 5.

assessment create next assessment	6
find person	5
assessment copy from previous	4
assessment medicines question	4
add edit person basic information	3
user change password	2
assessment diagnoses question	2
remove person	2
assessment kunta fin question	2
person pictorial report	2
assessment careplan notes	1
main support	1
person modify case	1
assessment form pdf	1

○ assessment medicine dra info	1
○ assessment careplan notes	1
○ dailynotes report	1
○ manage helptext	1

Raisoft application greatest strength	Raisoft application greatest weakness
Loogisesti etenevä arviointi	Yleinen sekavuus ja käyttäjälle tulisi olla näkyvissä vain ne asiat esim. raportit mitä tämä tarvitsee
Toimii pilvessä. Saa monipuolisia raportteja ja laatuanalyyseja.	Ohjelmiston käyttjien osaaminen ja sitä kautta arviointien luotettavuus. Arvioinneista saatavaa tietoa ei osata hyödyntää
Teknisesti toimiva, Raisoftin mittarikäsikirjat kehittuja. Laatumoduuli hyvä. Raporttien muokattavuus hyvä ominaisuus, joka nyt vasta asiakkaiden keskuudessa tiedostettu	Liian paljon vaihtoehtoja ja klikkauksia. Liian paljon raportteja. Liian monest paikasta pitää hakea tietoa
Arvioinnin muokkaaminen	Ei toimi offline. Päivityksen jälkeen liian usein esiintyy ongelmia
The assessment view has three main contents clearly: the question tree, the assessment questions and the manual	The report contents do not meet current THL specifications and it is difficult for users to understand how report results are generated (for example Organization statistics and Assessment count)
Greatest strength is its versatility	Greatest weakness is its versatility... There are so much our end users can do, but they do now know where to begin. I feel the way we display things in the application could be simpler and more intuitive. As an example, to see previous cases you need to go through "person and case information" - "show other cases" - and then you need to choose which one + write the reason why. When it also could be just 1 button beneath the active case "Show other cases" and a drop down list where you choose what case you want to view