Avrupa Bilim ve Teknoloji Dergisi
Sayı 35, S. 408-416, Nisan 2022

Araştırma Makalesi

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
No. 35, pp. 408-416, April 2022

Research Article

Examining the User Interface Development Stage in the Software Development Process

Bora Aslan[1*], Füsun Yavuzer Aslan[2]

[1*] Kırklareli University, Faculty of Engineering, Departmant of Software Engineering, Kırklareli, Turkey, (ORCID: 0000-0002-8069-8204), bora.aslan@klu.edu.tr
[2] Kırklareli University, Faculty of Engineering, Departmant of Software Engineering, Kırklareli, Turkey, (ORCID: 0000-0001-7096-3425), fusunyavuzer@klu.edu.tr

(First received 10 January 2022 and in final form 29 April 2022)

(**DOI:** 10.31590/ejosat.1055996)

ATIF/REFERENCE: Aslan, B. & Yavuzer Aslan, F. (2022). Examining the User Interface Development Stage in the Software Development Process. *European Journal of Science and Technology*, (35), 408-416.

Abstract

For the software development world, the simplest definition for interface design is designing websites, software, mobile applications or other informatics tools to create an interactive experience with the user for facilitated use. The purpose here is to meet the needs of the user, design the properties that enable the user to reach his/her targets and create a simple and visual representation of the data. Steve Job's expression "The design is not just what it looks like and feels like. The design is how it works." highlights that design must not be limited only to being aesthetically pleasing and created the bestseller products of today with various design awards. In this study, analyzes about user interface design were made throughout the software development process and tips were given for a successful design.

Keywords: User Interface, Software Engineering, Software Development Process.

Yazılım Geliştirme Sürecinde Kullanıcı Arayüzü Geliştirme Aşamasının İncelenmesi

Öz

Yazılım geliştirme dünyası için kullanıcı arayüz tasarımı en basit tanımıyla web sitelerinin, yazılımların, mobil uygulamaların veya bilişim araçlarının kullanımını kolaylaştırmak için kullanıcı ile etkileşimli bir deneyim sunma ortamlarını dizayn etmektir. Buradaki amaç; kullanıcının ihtiyaçlarını gidermek, kullanıcının amacına ulaşmasını sağlayan özellikleri tasarlamak, verinin basit ve görsel bir temsilini oluşturmaktır. Steve Job's, "tasarım, sadece nasıl göründüğü ve hissettirdiği değildir. Tasarım, nasıl çalıştığıdır" cümlesi ile tasarımın sadece estetik özelliklerin güzel olması ile sınırlanmaması gerektiğini vurgulayarak günümüzün tasarım ödüllü ve çok satanlar arasında yer alan ürünlerini ortaya çıkarmıştır. Bu çalışmada yazılım geliştime süreci boyunca kullanıcı arayüz tasarımı ile ilgili analizler yapılmış ve başarılı bir tasarım için ipuçları verilmiştir.

Anahtar Kelimeler: Kullanıcı Arayüzü, Yazılım Mühendisliği, Yazılım Geliştirme Süreci.

[*] Corresponding Author: bora.aslan@klu.edu.tr

1. Introduction

The most striking developments of the current century are related to the extraordinary expansion of information and communication technologies. Especially, developed software products bring solutions to individual or corporate needs in every aspect of life. It would be a mistake for public or private institutions only to consider these products as a means of faster service offering or resource management. The most important characteristic of the people in the 21st century is to deduce in life against changing technological developments, making conscious and well-informed decisions, applying knowledge to new situations, contributing to efficiency when generating new knowledge and sharing this knowledge and ensuring personal development. Therefore, all software developed for devices such as smartphones, personal desktop computers or laptops and tablets must meet the software need of the 21st-century individual. Surely, this need cannot only be considered business-centric. People mostly use these types of devices to access to internet only to meet their entertainment and communication needs.

One of the most important properties a software must have is to have a high positive impact on the user's work performance. Otherwise, regardless of how well-planned and well-coded the software is, the usability will be at a low level. Therefore, a strong background in software does not necessarily make it usable software. The preferability of the software with low usability will be at a low level. Generally, this function can be investigated under the software ergonomics title.

The most important component that determines whether a software is ergonomic is its interface. The performance development of the information systems has evolved from text interfaces to graphical interfaces for the software interfaces that we see today. Therefore, Graphical User Interface (GUI) concept comes into mind when we talk about interface today and this study will use the word interface with the meaning of GUI. GUI that enables interaction of the developed software system with the end-user directly impacts the usability and therefore, user performance.

2. Human-Computer Interaction and Interface Design

Human-computer interaction is a study field interested in interactive technology design, development and usability. In general, human-computer interaction is a field that aims for different disciplines such as human behaviour, psychology, information systems, graphic and industrial design, sociology, education science and software engineering to work coordinatively as in seen Figure 1. The most important purpose for the human-computer interaction field is to increase the quality of the relationship between humans and computers (Çağıltay, 2011).

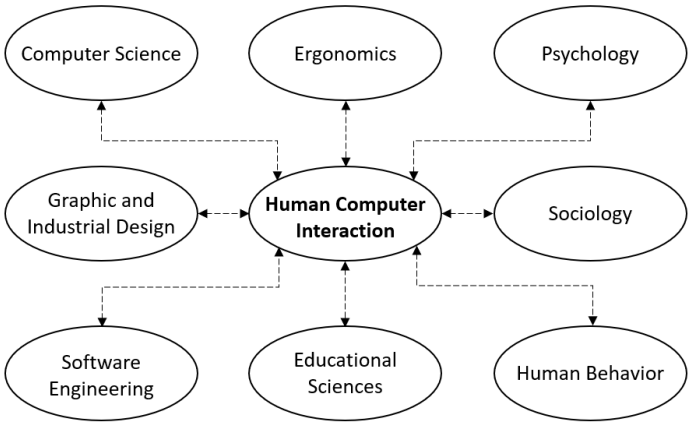

Figure 1: Human-Computer Interaction

One of the best examples in the history of the computer about this subject is the computer mouse which is the pioneers of human-computer interaction developed by Norwegian descent American inventor and engineer Douglas Engelbart (Pang, 2002). The general view and purpose of Engelbart were to increase the interactive experience by consolidating the interaction skills of the humans with the surrounding. Therefore, the computer mouse emerged among the most important computer peripheral units based on the idea that a computer can be controlled with better quality.

The human-computer interaction is not limited to traditional desktop computers. Such that, thinking computers only as desktop computers would be a highly narrow perspective in today's world. Tablets, smartphones, cameras, photographic devices, automobiles or all programmable devices that can interact with humans are included among the human-computer interaction field. Especially with the recently mentioned Internet of Things concept, the bigger picture can be seen more clearly as it is considered that all devices will be connected to the internet and therefore, interact with humans (Nielsen, 1994).

In the 1940s, computer technology was highly complex and at a level that only individuals with certain know-how can use it. In 1945 Vannevar Bush published an article called "As We, May Think" in Atlantic Monthly magazine (Bush, 1945). In this article, Bush mentions an imaginary machine called "memex". Memex screen is an imaginary machine with a table with a keyboard and selection buttons and records the data on microfilm. Of course, the irony here is that this imaginary machine express by Bush was going to be offered for human use in almost 50 years. Although memex was imaginary, it is clear that it played an important role in the design of today's computers. In 1962, American engineer and scientist Ivan Sutherland developed the first interactive graphic program of the Massachusetts Institute of Technology (MIT) called Sketchpad (Sutherland, 1964). This program that enabled drawing basic geometrical shapes on the computer by using a pen is accepted as the forerunner of CAD (Computer-Aided Design) software. Thus, different devices were added to the keyboard for interacting with a computer. In 1963, Douglas Engelbart invented the computer mouse and brought a different convenience to interaction (Pang, 2002). This was followed by different devices such as the joystick and lighten. However, the computer mouse preserves its position in today's world due to ease of use and low error rates. On the other hand, the complexity of the computers decreased and therefore, the usability increased in those years. Moreover, in 1968, Alan Kay

(designer of Smalltalk programming language which is the first object-oriented programming language) suggested the first personal computer for children called "Dynabook" when he was a doctorate student (Kay& Goldberg, 1988). This concept can be thought of as the primitive form of today's laptop and tablet computers. In the 1970s and 1980s, personal computer use expanded with Xerox Star workstation, Apple 1, Commodore Pet, IBM 5150 and LTE 286 (Abbott, 2017). Additionally, Xerox Star earned the title to be the first computer with a graphical user interface. Moreover, today's standard WIMP (window, icon, menu, pointing device) concept and WYSIWYG (What-You-See-Is-What-You-Get) were again started to be used in those years (Kim, 2015).

As the historical example shows, there are four main elements in human-computer interaction (Myers et all, 1996). The following four components should be considered in a coordinated manner and the system should be designed accordingly.

- User: All products (software or hardware) are produced for humans. Almost the only thing producers want is the effective use of their products. If we consider only software, the developed software must be designed according to user audience. Most of the day, software that addresses large user groups (web applications, office software etc.) are developed. Even in these situations, user needs should be taken into account. Prof. Yılmaz Taptık states that "all developed products and services are an offer to solve the customer problems." Therefore, the user is always the beginning of the solution cycle.
- Tool/interface: This is the structure the users use to do their desired task. Depending on the situation, a computer, tablet, smartphone or automobile can be accepted as an interface or a tool. On the other hand, software installed on the related hardware can be evaluated as a tool.
- Task: This is the job that the users do with the interface. A task that is accepted as the target for the human-computer interaction should be designed so that the users complete the tasks easily without struggling.
- Context: This is the environment in which the task occurs when the user completes it by using the tool. The context can be a physical environment such as a house, school or workplace or a virtual environment such as the internet, local network or social media network.

For example, when the navigation software in a smartphone is considered to find the directions to go to the post office, the user is the individual who uses the software. The tool is the navigation software itself. The task is the shortest and fastest directions for the user to reach the post office. The context is the environment of the user. The developed software should provide convenience to the user depending on the context. For example, if the navigation example automatically decreases its brightness when it is dark, the user will not experience any eye problems. At this point, it is important to note the fundamental principles of computer-human interaction.

The system must be designed with a user-centric approach. In the first years of computer technologies, there was system-oriented design because of having few users and system limitations. But today, a user-centric design approach is implemented not for software development alone but all product and service development stages. The products or services created otherwise will disappear. Although the user-centric design is considered as something easy, it is highly challenging. Especially for traditional software developers, understanding the user and designing for their needs and demands is often regarded as unnecessary. It is important to make large scale research about the users when an ideal software is designed. The age, gender, education level, social status, information technologies experience and even the cultural background of the user should be analysed properly. To give a simple example, a web application design for the 6-12 years old age group must be significantly different than a web application design for newspaper columns. Similarly, it might be important to follow the purchasing habits of shopping website members, recommending similar products and providing different interfaces for user genders (MacKenzie, 2012).

Most of the time, solutions are created by making various interviews by software for the designed system. In this case, the sociology and psychology dimensions of the human-computer interaction are highlighted. It is important to ask the right questions to the user to identify the task and understand the system correctly. This is directly linked to knowing the user. Depending on the task size and software engineering principles, the problem should be divided into sub-sections and processed and understood step by step. The task must be tried to discovered by interviewing as much use as possible.

The short-term memory of the individuals is the memory that can hold information for less than 30 seconds. It is proven that the short-term memory can hold a 7 unit (±2 unit) structure. Therefore, a normal individual can instantaneously store approximately 7-word sentences or 7-digit number. The designed system should function accordingly. For example, if a menu will be designed, the short-term memory of the user should be considered and the design should at most have 7-9 elements. It is important to remember that software must be easy to use and understandable to be memorable. To give another example, the number of buttons on the interface must be ideal. Otherwise, this might cause complexity on the user side.

For the long-term, the application must ensure consistency to ease the memory load of the user. It is beneficial for the users to work with familiar interfaces. For example, Office products developed by Microsoft such as Word, Excel, PowerPoint have the same interface even if all the products have separate functions. Consistency is an important property that connects the user to software.

In addition to defining the task, the error-free operation of the system is the most important success. Of course, it is ideal not to have any code errors and/or vulnerabilities when designing software. However, an error-free system design is not possible. In addition to that, the user should take the necessary instructions to use the system without making a mistake. Regardless of how well-coded the software is, there will be problems if the error potential of the users is not included in the design. Therefore, the system must prevent certain moves of the user within the system. For example, in the case of a shopping website, if the cart is empty, the purchasing movement of the user will be prevented so that the user does not complete purchasing. To give another example, the text box which will include the final grade of the student should accept values between 0 and 100. On the other hand, open and clear error feedback is important for human-computer interaction. For

example, instead of "an error occurred" warning, the software should have a specific message such as "the final grade should be between 0 and 100". To give another example, poorly designed error messages are often visible as messages with hex codes. In this case, the user does not understand the cause of the error. The end-user is interested in where the error is rather than the error code. The error message should be instructive i.e. it should inform about where do error occurs and what should be done. On the other hand, users should regularly get feedback. Feedback can be provided when things are well rather than when the error occurs. For example, most modern software provides various information during installation when everything is going well.

Especially in the last few years, the idea for software to more comfortably communicate with users became more important. Technology giants such as Microsoft, Apple and Google design their systems more sincere. In this case, it is better to use a more natural language that the user is accustomed to instead of the complex languages of the computer. Especially, the errors and operation steps should be presented with humorous and daily language. For example, messages such as "last 2 minutes to install" or "your system is configured at the moment" creates a natural environment for the user. Of course, software developed for different sectors must use different languages. For example, software developed for the health sector should be prepared according to the terminology of the doctors, nurses and other healthcare staff. The key point here is that the software developer should use the developer's language which is the computer language.

Users can easily get lost especially when they are navigating websites. The same is valid for desktop or mobile software. Regardless of which page or module the user is on, the user must always be presented with an option to go back to a familiar interface i.e. to the home page. In other words, the user must not be cornered. There might be certain exceptions when it comes to desktop or mobile software. But the escape points should be well-designed on websites. Otherwise, the user will want to use the "back" button on all web browsers and this might cause errors due to website properties. The escape points will help the users to easily navigate the system.

3. Usability and Software Ergonomics

Usability is a factor that directly impacts software quality. ISO 9241 defines usability as "target users completing the necessary tasks in an effective, efficient and satisfactory way" (ISO, 1998). The main targets of usability are given below.

- Effectiveness: This is completing the tasks by the users via software completely and accurately.
- Efficiency: The efficiency target is low resource use such as memory, disk and internet throughout the tasks.
- User Satisfaction: The developed software should be easily used by the user.

Usability is indispensable for a user-centric product development process (Spolsky, 2008). However, when it comes to the software world, this parameter is pushed to the background especially since there is time constrain to deliver the software. Yet, studies identified that software with high usability always has a higher product lifecycle. This awareness accelerated as usability is added to modern software development processes.

The human-computer interaction ergonomics is defined with TS EN ISO 9241-151 standard. This standard acts as a guide to developing interfaces that can be used by all users. The purpose of applying this standard is to identify how well a website meets the designated conditions and to improve the user experience in light of these results. This standard is a guidance document for high-level design decisions, design strategy, content design, on-site navigation, search and content presentation. More effective and efficient use is aimed at considering the behaviour patterns that the users are used to. On the other hand, ISO/IEC 40500:2012 standard sets the accessibility standards and criteria for website content. Accordingly, this standard explains how web content can be turned more accessible for disabled individuals. Usability encompasses a wide range of barriers such as visual, audial, physical, speech, cognitive, linguistic, learning and neurologic barriers. While this criterion covers a wide range of elements, it does not meet all the needs of the disabled in terms of disability type, level and combination. At the same time, these criteria make web content more suitable for elders which changing skills due to old age and generally increase the usability for most users.

Another important aspect is the software usability tests. Software that successfully completes these tests are accepted as ergonomic. While these tests can be conducted at laboratory settings where real users test the developed software, they can also be conducted by sending the software to selected users and receiving feedback from these users (Dumas, 2008). Some of the usability tests applied today are given below:

- *Eye-tracking test:* This test method is often applied with various special devices in a laboratory. Where and how long a user looks on the screen and focuses are identified and reported. This way the hot spots which attract the most attention and cold spots which attract the least attention can be identified. This test uses a device that tracks the user's eye. The eye movement of the interface is transferred to the computer via a transmitter and the areas with high eye movement on the screen are marked as hot spots. In our country, universities that work on human-computer interaction conduct these types of tests. Figure 2 shows the eye-tracking system.

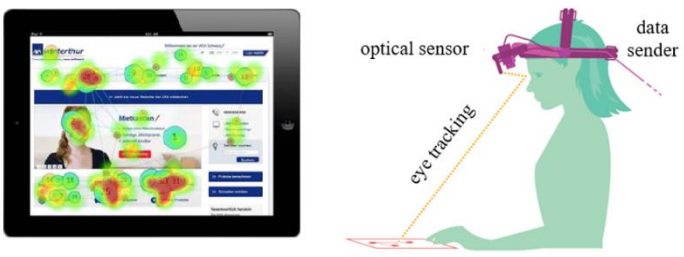

Figure 2: Eye Tracking Test

- *Guerrilla test:* This test method is often applied to randomly selected test users outside the laboratory settings. During the test, user views, movements and facial impressions are recorded. This test method which is often applied online at the internet setting is completed with the feedback collected from the users.

Various game developers test their products by using this method. While the test results aim to identify the user expectations and interest, differences can be identified for demographic information such as age, gender and where the users live. In this technique, the Google Analytics tool is frequently used especially for web applications. With Google Analytics, different tests such as real-time reporting, content analysis, speed analysis, social media analysis and ad analysis can be conducted.

- *Expert Analysis:* In this test method, the developed application is tested and reported by the experts. Tests conducted by experts can significantly determine usability rates. Using their acquired skills and experience, usability testers and professionals evaluate applications and audit all relevant parameters.

4. User Interface Design

All around the world, people are clicking, tapping, dragging, typing and touching on glass screens thanks to computer and internet technologies. Computer interfaces which only had a keyboard and keyboard-like keys at the beginning consist of glass screens and the future might evet bring holograms.

If a platform-based classification is applied for the software developed today, the most logical classification would be desktop, web and mobile software. Of course, the new technological advancements are eliminating the barriers between this classification. Moreover, desktop software is decreasing every day and the necessity for such software is questioned. Even the most popular office software is now offering services on the web. Therefore, some experts emphasise that desktop software development is unnecessary especially for the end-user and web and mobile software development should be highlighted. Such that today, almost all software solutions that meet the needs of the institutions offer services on the web environment and can be easily sued with a simple web browser.

Since each platform has hardware or software limitations and opportunities of its own, they have different user interface development styles. These operations often designed as GUI i.e. graphic user interface does not have a standard. In fact, it is generally challenging to talk about good design when there are standards.

4.1 User Interface Design for Web Applications

Especially almost all of the desktop automation software is replaced by web systems. WWW (World Wide Web) developed by Tim Berners-Lee in 1989 has significantly developed and evolved until today. This is the end of web 2.0 which evolved from web 1.0 where the user was passive to an active user who creates and shares content and has a word to say. Web 3.0 also known as the semantic web offers a highly customizable environment where artificial intelligence is highlighted and where search engines interpret the user-generated content. In addition to the advanced underlying technology, the design is different and advanced according to today's requirements and opportunities (Bevan,2005). In general, the user interface design of the web systems contains 4 elements.

Layout and website structure: The layout is a template that expresses how the website pages should be, where the logo and other necessary graphics, navigation menus and ad spaces should be placed. In general, the layout clearly designates the header, navigation system, main content and footer sections. The layout should be planned so that the user can easily read and follow the page. To determine the places of these sections, HTML5 tags such as <header>, <footer>, <nav>, <main>, <section> are used in the coding section. Website structure is the hierarchy or website map. Thus, all pages can be seen as a list.

Font and typography: Canadian font designer Robert Bringhurst defines typography as the art of turning the spoken language into a long-term visual format. The font for the web applications should be selected to be easily readable and suitable for the design. Surely, readability should be the main priority. When a correctly used typology meets the design, the impact significantly increases. However, it is important to remember that the primary purpose of typography is conveying the written information. On the other hand, using various fonts in the web environment can create problems for different web browsers. Although this problem is mostly overcome, fonts on Google (https://fonts.google.com/) can be preferred to avoid problems. It is important to use typography correctly. For example, using different fonts in a paragraph, too narrow or too large line spacing have a directly negative effect on the readability.

Colours: It is important to be careful about website colour selection. Although there are unlimited colour options, colourful websites have always been disturbing. Designers often recommend creating a design at most with 3 colours. As the number of colours on the interface decreases, the consistency increases and the messages that will be delivered to the user can be conveyed simpler and more clearly. Certain tools on the website can be used for colour selection. https://color.adobe.com/ and http://www.paletton.com/ are only some of these tools. Each of the 16.8 million colours available for the web environment is represented with a code between #000000 (black) and #FFFFFF (white). It is important to choose a suitable colour palette for web applications. Therefore, it is necessary to get support from professional tools and individuals. It is known that some colours have an impact on human psychology. For example, red colours invoke excitement and action and also represents power, importance and youth. The yellow colour will help individuals to relax and gives an optimistic atmosphere. The blue colour is ideal for seriousness, freedom and a pleasant look and it is the most common option. The orange colour will have a warm feeling. White colour represents simplicity. The green colour symbolises peace and health. Black colour is about passion and power and works well on websites. Black is especially preferred for designing websites about jewellery or luxury cars. Grey is a neutral colour and especially used for representing gaps. Additionally, simplicity and balance in most designs are achieved with grey colour. White represents cleanness and simplicity. It is a complementary colour as it can be used with all colours. Of course, these explanations are the generalized form of the colours. Different colours can have different impacts on different cultures. For example, while white represents peace and marriage in the European culture, the same colour represents mourning in the Japanese culture and death in Chinese culture. Therefore, the assumptions above can be disregarded especially for global designs such as website designs.

Images and graphics: It is important to make sure that the graphics and images are optimized. For that purpose, certain graphic programs such as Photoshop and Gimp can be used. It is

important not to use too large or too small images. The image size should be directly applied to the interface design. If possible, online image and graphic banks should be used for images and graphics. The licences for images and graphics selected online should be checked. For free solutions, images and graphics with CC (Creative Commons) licence can be used.

The above-mentioned user interface design elements for our applications turn into meaningful and functional applications when applied with the right technologies. In general, designs often reach software developers after the expert designers complete their process. Moreover, various websites sell templates online. The first duty of the software developer is to re-organize the design with certain development tools and turn the design into a programmable format (Ferré, 2001).

One of the main problems for web interfaces is the resolution. Each device supports a different screen resolution. While desktop or laptop computers can work with high resolution, smartphone screens naturally have lower resolution. It is desirable for web applications to effectively work on all devices. Therefore, all web interfaces today must be responsive for different devices. Responsive design is about giving an interface that does not shift or disrupt different devices. Today, various responsive libraries have been presented to the software developers by using HTML5, CSS and JavaScript. It is easy to create responsive designs by using libraries such as Bootstrap, Foundation and Skeleton. In a near future, it is expected that this type of design technique will be a standard web application. Therefore, it is recommended to test all designs on different browsers and devices.

4.2 User Interface Design for Mobile Applications

The advanced technology and opportunities have significantly developed the world of tablet and smartphone. The most exciting application development area in today's world is the mobile world. It is important to know the environment well to develop applications for this world that can be controlled with glass screens without a keyboard and mouse. On the other hand, it won't be wrong to suggest that the emotional attachment between any technology developed in the world and humans is not as high as our attachment with mobile devices. Because mobile devices have the indispensable applications of the 21st century such as clock, music player, phonebook, notebook, newspaper and navigation device. Of course, social media is the key to such emotional attachment. Such an important environment has become the key focus for software developers. Additionally, the human interactions by using the sensors of these devices eliminate the borders of software.

As it is known, the mobile world is known for its operating systems rather than the product brands. Currently, the most commonly used operating systems are iOS, Android and Windows Phone. Unfortunately, mobile applications are developed for platforms. Although stable, platform-independent application efforts continue, an application developed for one platform cannot operate on the other. Therefore, developers first need to select a platform for the product. Each platform has different user interface design options.

Platforms often publish their development opportunities on websites. Platform developers continuously publish the new versions of the operating systems based on the technological developers. Of course, these releases are continuously followed by developers and designers. Related development opportunities for 2 important platforms are iOS and Android.

It would be wrong to regard mobile devices as the smaller versions of desktop or laptop computers. These devices have their own limitations and properties. For example, various properties and limitations such as small screen size, low performance, battery life, screen interaction area and touchscreen property, directing (vertical and horizontal rotate) guide the developers to different directions than desktop software development techniques. The following question might come into mind. If responsible web applications can operate on all types of platforms, why do we need mobile applications? Actually, the answer to this question can be seen from the above-mentioned information. Mobile devices are slower than desktop devices. Additionally, web applications are often prepared for mouse and keyboard use and need a tool for working. This mediator is the web browser. Even though the application is responsive, web browsers make the texts, images and similar web elements smaller and shift these elements which cause the mobile device to run the application slowly. On the other hand, mobile devices interact with humans with touch and finger movements while web applications are designed for mouse and keyboard. Of course, various problems are due to differences in coding standards. Scripts in web applications are generally prepared for desktop web browsers. Therefore, it is natural for some scripts not to work on mobile browsers. Another thing here is the mobile website approach which was popular in the past but started to become unpopular today. In this type of approach, an additional mobile interface of a web application is created and launched on a different web address. The user that visits the related webpage from the desktop computer and the user that visits the same webpage on the mobile device accesses different interfaces. In this case, an interface optimized for a mobile device is presented to the user. Although it seems like a good solution, not being able to use the interaction presented by the mobile device entirely, lack of integration with other applications and using a browser has negative impacts on the performance. Additionally, the customizability of the mobile web approach is low.

When the interfaces of mobile applications are prepared, they must be tested on real devices. Although the developed app looks beautiful on the computer screen, it might look distorted on the mobile device. The font should be selected carefully. The small screen size of the mobile devices can cause some fonts to look thin. Similarly, the right graphics should be selected and the graphics should be optimized for mobile devices. It is important to be careful about animation use. Most mobile device users are impatient. Since unnecessary animations will delay the app launch, the users will feel that the app is running slow. If we list the top 10 things the 21st-century individuals don't like, slow devices and applications would be on the top.

It is necessary to avoid fixed areas to navigate inside the application. All web browsers eliminate menu and some toolbars to increase the user navigation area. Similarly, fixed menus are not used on applications developed for mobile devices. If a menu is necessary for an application, the hamburger menu with three lines as the symbol should be used. The hamburger menu is loved by the menu users and this menu became a standard for Android mobile software and all front-end frameworks.

5. User Interface Development Process

Most software developers accept that usability is among the most important parameter in today's world. Accordingly, the importance of an effective and usable interface development cannot be mistaken. This section will talk about the stages of the user interface development process.

5.1 Creative Design Workshop

Leaving the software interface to the software developer alone is one of the greatest mistakes. Because software developers are often software or computer engineers and they often lack visual creativity skills. Therefore, a workshop that gathers target audience, software developer, designer and business owners can ensure a better user interface. The software developer can choose the development environment and opportunities, the designer can choose the colours, design elements and social impact and the user and business owner can choose the expectations and desires from the software. These actors can be gathered in the software development process to take great steps. Of course, the number of individuals participating in the workshop should be selected carefully. Some sources state that the number of individuals shouldn't exceed 12. Large group work can be done among the participants. Moreover, large international software development companies can even organize the workshops by giving scissors, papers and coloured pens to do a handicraft as in seen Figure 3. The main purpose here is to increase the communication between the user, software developer, designer and business owner. At the end of the workshop, the usage purpose of the software as well as who it addresses and how it will achieve the purpose will be found. Additionally, the software platform should also be discussed in the workshop. For example, if a mobile application will be developed alone, the software developer should introduce the platform and talk more about the opportunities rather than the limitations.

The studies have shown that this type of workshops is effective. Especially, designing the application by cutting from a paper and pasting which does not involve computer use helps the borders between the end-user and software developer to collapse. This method reveals different perspectives for the software interface. Including an actor with a designer identity in the process highly increases the software value. Over the years, the colours and elements that the individuals want to see from the software change. This is especially more visible in the mobile world. Therefore, an expert designer in software interfaces can directly impact the usability of the software. As all actors involved in the process, the ownership of the developed software will be high.

Figure 3: Creative Design Workshop

5.2 Creating and using personas

It will be easier to develop software when the developer empathises with the users. Therefore, user personas must be created. This name comes from the Latin word persona. Creating and using personas is critical for a user-centric software engineer. This way the identity of the users can be revealed. Generally, the simplest definition for a usable software demand is "my mother should be able to use it". However, the software design principles cannot be simplified this much. Thus, the usability of the software is not directly proportional to ease of use. In this case, the user should be known as best as possible. At this point, persona card or personality card intervenes. There is a short profile of the individual on this card. User's name, surname, photo, job description, personal and occupational characteristics, authorities and similar information are on this card. Some software developers see the users as desktop computers but this is a mistake for end-user-centric software engineering. Generally, persona cards are not designed in fixed forms and include humous elements (Figure 4).

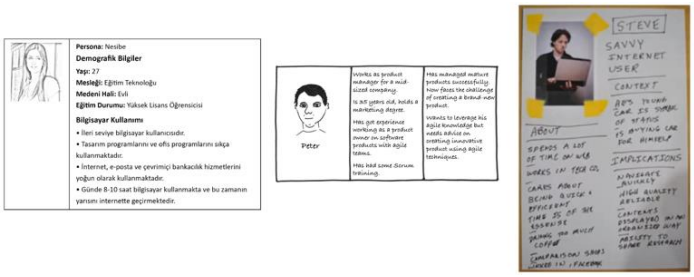

Figure 4: Persona Samples

5.3 Creating and using personas

If the coding process starts with a mistake or undesired way for the user in the software interface, the codes need to be re-written and significantly change to go back to the interface design. Therefore, the interface design should be sketched first and then, the coding should start later. In this case, design is achievable at three stages as in Figure 5.

- Sketch: This is the interface design only on paper to give a broad idea. This way it is possible to determine where the elements that will be added to the software will be placed. There is no need to sketch with the exact size. Generally, it is prepared in 3-5 minutes without too much attention. The purpose of this is to give a broad idea about the application interface. It is possible to make changes or corrections if something is not liked.
- Wireframe: This is preparing a sketch with almost exact dimensions and clear lines via a computer program. The elements are clearly shown. This is the visual template of the design. Generally, sketching software such as Photoshop, Illustrator, MS Visio is used for preparation. However, there is a special software for this purpose in today's world. For example, Edraw Max is developed to create a user interface template for iPhone. In addition to that, there are web-based solutions such as mock flow, Pidoco (https://pidoco.com/) and Balsamiq (https://balsamiq.com/).
- Mock-up: This is the software design closest to the real software. After agreeing on the template, this is created by either the software developer or designer with the

help of an editor. If there are not many structural problems, the mock-up is final and the software is coded for this interface. Of course, a software developer must be involved in this process and highlight the limitations. For example, if the software will be developed or a low capacity platform such as a hand terminal, the developer should note that 3-dimensional images or animations must not be used abundantly.

Figure 5: Sketch, Wireframe and Mock-up

If these three stages work well and if the parties reach an agreement, the time on software design will significantly decrease. Of course, these three steps must be applied for all possible interfaces of the software and a flow diagram should be created. This diagram must have the connections of different interfaces of the software and a larger picture should be painted. The escape points mentioned before can be easily seen on the flow diagram as in Figure 6 and the points where the user is trapped can be identified.

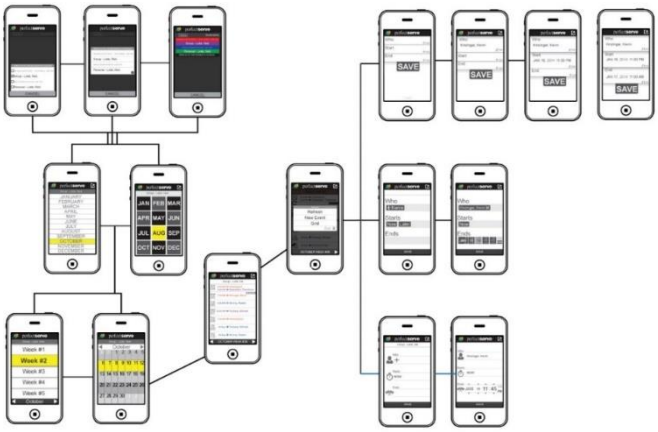

Figure 6: Interface and Structure of a Mobile App

As it can be seen, the stability which is the agreement between the user and developer increases throughout the process. This process must have continuous feedback. With this feedback, it is possible to make the necessary corrections easily when the application design is at the template stage. The mock-up stage is where the process becomes sensitive which draws the lines of the application and the match with the real application level increases at the stage. At this stage, the feedback decreased and there is an agreement with the user. The final design which is the last stage of the application is the saturation of the compatibility and the feedback is minimized. The next stage is the coding stage.

6. Conclusions and Recommendations

Today, the software development process differs significantly. It is almost impossible to develop an application independent of the user experience. Therefore, the user-centric software engineering concept is highlighted.

The only area where the user interacts with the computer in terms of software is the user interface. Therefore, interface design has an important position in the software development process. Interface design is about the human-computer interaction space. It is important to develop interfaces by knowing humans, therefore users. Therefore, users should be included in the development process. Organizing workshops are among the main tasks of software developers and designers to have matured ideas based on user views. It is possible to get to know the user and identify the needs of these workshops.

In interface design, internalizing the sketch, template and mock-up stages until the final design and receiving lots of feedback in the meantime will help saving time to complete the software within the designated time. Going back to design after the coding stage takes the entire process to an irrecoverable and challenging stage. Therefore, a significant amount of time should be allocated to software development for interface design.

Social media integration should be used for user membership. Increased password complexity is among the major problems of today's individuals with an increased number of system and web applications. Therefore, various web or mobile applications offer different options to users when creating a membership. For example, creating a membership with Facebook or Gmail accounts has become popular. Thus, the users can use multiple applications with a single membership. In addition to that, it is possible to pay with Google Wallet or PayPal for electronic commerce payment operations. Even if these options are not used by the user, it is important to provide these opportunities. Additionally, well-known brands such as Facebook and Google can introduce an added value to the developed application.

Just like it is impossible for someone who does not read a book to write a book, it is equally impossible for someone who does not look into the applications to design an application. According to 2020 statistics, there are more than 2 million apps on App Store and more than 2.8 million apps on Google Play. A good designer and application developer should continuously monitor new applications, new developments and trends. By following the success of websites and applications that are appreciated by the users and inspiring by them is among the rules of good application development.

An image bank must be created for the graphical elements that will be used in the developed interface. Icons are especially indispensable for web application or web interfaces. How metaphors can be provided by icons or images is explained in the first sections. For example, a home icon can be used for a home page. Of course, re-designing an icon or turning it into a suitable design instead of copying a website will increase the uniqueness.

In graphical interface design, buttons with functions such as add, change, send or save must be designed with high contrast (for example, blue background, white text) and buttons with negative functions such as cancel, delete, reset must be designed

with low contrast (for example, transparent background, black text). However, the negative function must have high contrast (for example, red background, white text) when a negative function is next to a neutral function. For example, delete and cancel functions should be considered accordingly.

One of the new approached in application design is to present the feedback on a region on the application screen rather than pop-up messages. Generally, a small area at the bottom of the desktop software is allocated for this function. If an error occurs, the related message is shown in this region with a red colour. A successful situation such as adding a record is shown with a green message.

Complex interfaces must never be presented to users. Using 30-40 textboxes and 15-20 buttons on an interface will make things challenging for the user. The user interface should be designed with a “less is more” approach. If carefully evaluated, the user membership number significantly increase by using different stepwise interfaces rather than long text boxes that were used in the past. It is important to remember that simplicity is glamour.

References

Pang, A. S. K. (2002). The making of the mouse. American Heritage of Invention and Technology, 17(3), 48-54.

Bush, V. (1945). As we may think. The atlantic monthly, 176(1), 101-108.

Sutherland, I. E. (1964). Sketchpad a man-machine graphical communication system. Simulation, 2(5), R-3.

Kay, A., & Goldberg, A. (1988). The dynabook: past, present, and future. In A history of personal workstations (pp. 249-264).

Abbott, R. (2017). A study of the growth and evolution of personal computer devices throughout the pc age (Doctoral dissertation, Cardiff Metropolitan University).

Myers, B., Hollan, J., Cruz, I., Bryson, S., Bulterman, D., Catarci, T., ... & Ioannidis, Y. (1996). Strategic directions in human-computer interaction. ACM Computing Surveys (CSUR), 28(4), 794-809.

Nielsen, J. (1994). Usability engineering. Elsevier.

ISO (1998) International Standard Organization: ISO 9241-11(Guidance on usability), International Standard: ISO 9241-11(Guidance on usability)

MacKenzie, I. S. (2012). Human-computer interaction: An empirical research perspective. Newnes.

Kim, G. J. (2015). Human-Computer Interaction: Fundamentals and Practice. CRC Press.

Dumas, J. S., & Loring, B. A. (2008). Moderating usability tests: Principles and practices for interacting. Morgan Kaufmann.

Ferré, X., Juristo, N., Windl, H., & Constantine, L. (2001). Usability basics for software developers. IEEE software, 18(1), 22.

Spolsky, A. J. (2008). User interface design for programmers. Apress.

Çağıltay, K. (2011). İnsan bilgisayar etkileşimi ve kullanılabilirlik mühendisliği: Teoriden pratiğe. ODTÜ Geliştirme Vakfı Yayıncılık.

Bevan, N. (2005, July). Guidelines and standards for web usability. In Proceedings of HCI International (Vol. 2005).