Survey on Well-known User-interface Design Rules

Amir Rastegar Lari

Department of Computer Science, University of Porto, Porto, Portugal

Abstract— In this article, we investigate some well-known User-Interface (UI) design rules. User interface design or user interface engineering is the design of user interfaces for machines and software, such as computers, home appliances, mobile devices, and other electronic devices, with the focus on maximizing usability and the user experience. User-Interface design is not a straightforward process. Design rules often are based on goals in compare to actions. Usually, it is not possible to satisfy all the goals, a tradeoff is needed. Most of the rules are based on users and their goals. Some of them use funny approaches like gamification to engage using application more. Designing user-interface well is the main approach in developing a system and an application in Human-Computer interaction field and to assess how easy user interface design is to use, usability testing could be applied.

Keywords- User-Interface Design, Users, Tasks

I. Introduction

The goal of user interface design(UI) is to design user interfaces for different software platforms such as mobile applications, desktop software, and web applications with the aim of maximizing User-Experience (UX), usability and user engagement. UX refers to a person's emotions and attitudes about using a particular product, system or service. It includes the practical, experiential, effective, meaningful and valuable aspects of human-computer interaction and product ownership. Additionally, it includes a person’s perceptions of system aspects such as utility, ease of use and efficiency. User experience may be considered subjective in nature to the degree that it is about individual perception and thought with respect to the system. User experience is dynamic as it is constantly modified over time due to changing usage circumstances and changes to individual systems as well as the wider usage context in which they can be found. In the end, the user experience is about how the user interacts with and experiences the product. Usability is the ease of use and learnability of a human-made object such as a tool or device.[1]

In software engineering, usability is the degree to which software can be used by specified consumers to achieve quantified objectives with effectiveness, efficiency, and satisfaction in a quantified context of use.[2]

The object of use can be a software application, website, book, tool, machine, process, vehicle, or anything a human interacts with. A usability study may be conducted as a primary job function by a usability analyst or as a secondary job function by designers, technical writers, marketing personnel, and others. It is widely used in consumer electronics, communication, and knowledge transfer objects (such as a cookbook, a document or online help) and mechanical objects such as a door handle or a hammer. Usability includes methods of measuring usability, such as needs analysis and the study of the principles behind an object's perceived efficiency or elegance. In human-computer interaction and computer science, usability studies the elegance and clarity with which the interaction with a computer program or a web site (web usability) is designed. Usability considers user satisfaction and utility as quality components, and aims to improve user experience through iterative design.

The main objective of user interface design is to simple human computer interaction that user could easily achieve the goal easily with minimum waste of effort. In the good user interface design, user could do the job without noticing software. To make better usability, different factors could be considered such as typography and graphic design. These could affect on when the user is doing certain interaction and could make better or worse the potential of the user to do certain interaction. With the aim of having a usable system that could addapt to user needs, two factors such as psychology visual elements and technical functionality should be considered. As user interface design involve different platforms, disigners should be expertise and have enough skills in their platforms. Also, in user interface design, user needs should be understand well. Each user interface design could have several steps and processes that some of them may be related to each other [1][2].

In this paper, we are going to review some well-known user interface design rules be used by most companies.

II. BEN SHNEIDERMAN UI RULES

In 1987, Ben Shneiderman, distinguished professor in university of maryland, introduced eight golden rules for designing productive and disappointment free user interface. Apple, Google and Microsoft are such as companies that used shneiderman UI rules for their successful products. Considering these eight rules while designing could help to have better design.[3]

A. Strive for consistency

In order to design consistent interface, the same design patterns and the same sequences of actions should be used for similar situations. Using the familiar icons, menu hierarchy, right color, typography, terminology in prompt screens, commands could be included but not limited. In this way, users could do their goals more easy and they do not nead to learn new representation of the same action with consistent interface.[3]

B. Enable frequent users to use shortcuts

It is needed for quicker methods when using the systems becomes more. Using UI rules as shortcuts will help them to get advantage of shortcuts. In particular, it could be more useful when it is needed to do the same job more often. Such shortcuts like Abbreviations, Function Keys, Hidden Commands, Macro Facilities could be useful for expert users. As an example, In Windows and Mac users could use keyboard shortcuts for copying and pasting, so as the user becomes more experienced, they can interact with the user interface more quickly and automatically as they get more experience.[3]

C. Offer informative feedback

There should be some system feedback for every user action in reasonable amount of time. The feedback should be understood easily by human and the user should understand where they are and what is happening. For usuall and minor actions, the response can be selfconsious, while for unusuall and main actions, the response should be more significant. As a bad example when we often face with an error code instead of a human-readable and relevant message[3].

D. Design dialog to yield closure

It should be used beginning, middle, and end to organize sequence of actions. Users should not guess their actions. They should know what their actions will led to them. By completing each group of actions some feeback should be sent to users. This could provide a satisfaction of accomplishment. The informative feedback at the completion of a group of actions gives the users a satisfaction of accomplishment with sense of relief and a signal to drop possibility plans. It also shows that the way for next group of actions is clear. For example, users will receive a “Thank You” message and a proof of purchase receipt when they’ve purchased online.[3]

E. Offer simple error handling

Try to design a system with least user interface errors in order to avoiding the user for making a serious error.

Generally, no one likes to be told with errors. In case of error happening, the system should detect the error and offer simple, step by step uderstandable mechanisms for handling the error quickly and painlessly. For example, when the users forgot to provide input in an online form, bold the text fields.[3]

F. Permit easy reversal of actions

The designers should offer users easy and obvious way to reverse their actions. This feature is about feeling of worry. As the user knows that errors can be undone, this encourages them seeking of unfamiliar options. Reversibility could be some single actions, a data entry, or a complete group of actions.[3]

G. Support internal locus of control

Charging the system is strongly desirable for experienced users that they are initiator of actions and that the system responds to their actions in full control of events. Making users the initiators of actions rather than the responders should be considered in design systems.[3]

H. Reduce short–term memory load

The human attention is limited and could keep a small items in short-term memory at a while. Information processing in short-term memory requires that interfaces be kept simple with proper information hierarchy. Multiple page displays be consolidated, window-motion frequency be reduced, and sufficient training time be allotted for codes and sequences of actions.[3]

Specific client needs will dictate how to prioritize the usage of these rules. The client should never be taken out of the equation, and it might be the case that one or more of these rules results in superfluous and obtrusive design (as well as an additional development expense). Where simplicity is one of the main guiding principles in UI design, we should not make use of more guidelines than are necessary to successfully accomplish our task. This is, of course, not to discard the importance of these rules, but rather to reconsider their usage contextually; this will ensure that they are applied toward the success of a design, and not otherwise.

Specifically, let's take as an example rule B. While enabling frequent users to use shortcuts may be helpful for some desktop applications, this would almost certainly not be the case on the application's smartphone counterpart. Yet another example, this time considering rule 8. Advanced applications for power-users would need a vast number of features and therefore a more complex interface; although we would be adding to the short-term memory load, in this particular case, such a design decision might be appropriate.[3]

III. NIELSEN NORMAN GROUP USER INTERFACE DESIGN RULES

Nielsen Norman group is one of the world leaders group in UI and UX. In 1995, they introduced ten usability heuristics for user interface design. It has ten principles for interaction design. They are called heuristics because they are not specific guidelines for usability and expectation, they are broad rules of thumb. Their heuristics are used in many companies products such as Adobe, Apple, and Google and they could improve usability, utility, and desirability of designs.[4][5]

A. Visibility of system status

The users should always know about what it is going on the system by some feedback through the system inappropriate time. They should easily understand the system status on screen with a reasonable amount of time.[4][5]

B. Match between system and real world

Designers should endeavor the users' language which used for speaking with the system. Concepts, words, and phrases should be easily understood by the user, rather than system-oriented terms. Making information should appear in a natural and logical order as a real-world convention. This will make systems easier to use.[4][5]

C. User control and Freedom

Users should easily control their functions and have enough freedom to leave unorder state without obligating to through a complicated dialogue. Backward steps should be considered such as undo and redo as the users usually do mistake when working with systems.[4][5]

D. Error prevention

It is better to design carefully to prevent errors in the first place rather than good error messages. Checking error-prone condition and remove or control them to present users with a confirmation option before they commit to the action. The potential errors should be kept minimum because users are not interested to detect and remedy problems. Also, flagging actions could be considered as a mean of error prevention.[4][5]

E. Recognition rather than recall

Usually, recognition is easier for users than recall. The cognitive load should be at least by maintaining task-relevant information within the display while users explore the interface. It should be tried to make objects, actions, and options visible to minimize the user's memory load. The user should not have to memorize information from different parts to another. Using of the system should be visible or easily retrievable whenever appropriate this is because of limitation on human attention. We could only maintain around five items in our short-term memory at once. As a result, designers should be certain that users can simply employ recognize instead of recall. For example, doing a test on multiple choices is easier for us to answer questions on a test because it just needs to recognize the answer rather than recall it from our memory.[4][5]

F. Flexibility and efficiency of use

As using system increase, it is demanded faster navigation and less interaction. For this aim, some issues such as abbreviation, hidden commands and function keys could be used. Accelerators that new users have not seen may help to speed up the interaction for the expert user such that the system can provide to both inexperienced and experienced users. It assists users to do frequent actions.[4][5]

G. Aesthetic and minimalist design

Dialogues should include useful information and irrelevant or rarely needed information should not be used. Extra information in a dialogue participates with the relevant units of information and decreases their relative visibility. Unrelated information should be minimal because they will consume a user's limited attentional resources, which could inhibit a user's memory retrieval of relevant information. As a result, the display must include only the necessary components for the current tasks, whilst providing clearly visible and meaningful of navigating to other content.[4][5]

H. Help users recognize, diagnose, and recover from errors.

Designers should assume users could not understand technical terminology, therefore, Error messages should almost always be expressed in plain language to ensure nothing gets lost in translation.[4][5]

I. Provide online documentation and help

It could be ideal if the users could use the system without any helping or documentation. However, it may be needed to provide some help and documentation. The helping system should be easily searched, focused on the user's tasks, define steps for doing tasks and not to be confused and large.[4][5]

IV. JOHNSON USER INTERFACE DESIGN

A. Focus on the users and their tasks, not on the technology

When designing, it is critical to understand the users and their needs. The tasks that they want to do with the system should be considered. Another important issue is the context in which the software will function. It should not mostly focus on technology rather than the user.[6]

B. Conform to the user'*s view of the task*

In designing, try to strive for naturalness and use users' vocabulary not your own as a designer. Program internal must be concealed from the user view inside the program. Also, try the correct point on the power/complexity tradeoff and user should conform to view the task easily.[6]

C. Design for the common case

Common results should be easily achieved by each user. Usually , there are two types of common, more often and less often. When designing, it is important to design for core cases and does not sweat edge cases.[6]

D. Do not complicate the user'*s task*

The user task should be simple and could be easily understood. Don't give users extra and sophisticated problems and do not make users reason by elimination.[6]

E. Facilitate learning

In design, provide a low-risk environment that should be consistency. Provide an environment by designing that user could easily learn and interact less.[6]

F. Deliver information, not just data

The design should display carefully and if needed get some professional help from others. The screen belongs to the user to provide information, not just data and preserve display inertia.[6]

G. Design for responsiveness

Every task that the user do, acknowledge user actions instantly. It could be helpful to provide some way in order to let users know when software is busy and when it is not. While users are waiting for doing some part, free users to do other things. The movement should be animated smoothly and clearly. Users could be allowed to leave lengthy operations they don't want an estimate how much time operations will take. Letting users set their own workspace could be helpful.[6]

H. Try it out on users then fix it

Test your design result on users. Sometimes it could surprise even experienced designers. The designer should schedule a time to correct problems found by tests. Usually, testing has two goals: informational and social. Every time and purpose could have different tests.[6]

V. GAMIFICATION IN USER INTERFACE DESIGN

Gamification is the application of game-design elements and game principles in non-game contexts to improve user engagement, organizational productivity, flow, learning, crowdsourcing, employee recruitment and evaluation, ease of use, and more. The key here is to use elements of game design (fun, motivation, reward) to get users to do something that is in their benefit (and deepens your business goals). LinkedIn pioneered the UI pattern of profile completeness (and just-in-time tips and prompts) as a way to prompt users to share more information. This is now a default design pattern in web applications.[7][8][9]

Gamification, as a recent phenomenon for UI and UX designers , is a powerful approach for designing UI in order to increase user engagement and good experience. Firstly, you use this tool to enter fun elements in UI applications and systems. Users face challenges, whether challenging themselves or trying to win awards. Generally, they enjoy it. Secondly, the dynamics designers incorporate in successful gamification serve as effective intrinsic motivation, themselves – meaning users engage with the system because they want to. For instance, Foursquare/Swarm promotes users to “Mayors” of establishments after so many visits, enabling them to vie for a top place while enjoying meals, shopping, movies, etc.[7][8][9]

There are a variety of gamification techniques that can be considered in user interfaces, alongside other fundamental observations of what makes a gaming experience enjoyable & rewarding. By designing a product with these themes in mind, the aim is to create positive experiences that hook users to your product. Some of the gamification techniques for UI design are:

A.Badges ⁄ Medals

Badges or Medals are a simple way of providing feedback to your user while bringing a sense of reward for completing a task. This reward creates motivation to continue and can also help offset any negative experiences during the task itself.[7][8][9]

B. Delightful interactions ⁄ instant gratification

Adding small pieces of magic throughout your product makes the experience fun and enjoyable to interact with. The core functionality of the application must be reliable and usable before elements of delight can be added, as an interface that is unreliable or does not work as expected will create feelings of frustration and friction that overpower any touch of delight or fun.[7][8][9]

C. Fluid navigation ⁄ experience

When navigation of video game user interfaces is done right, they bring fluid and smooth experience to the game, and when done perfectly — bring no friction to the players’ experience. Whether you need to switch items, update an option setting, or simply pause the game — when nothing disrupts the user the experience is seamless. Bringing this to product design translates to providing fluid navigation of your product, where transitions between pages & forms, for example, should give the user clear feedback & understanding of what is happening at all times.[7][8][9]

D. Leaderboards

Depending on your product, leaderboards could be a feature to consider adding a sense of competition for your users, where you incentivize them to compete with each other, to increase engagement and fulfillment of completing tasks, etc.[7][8][9]

E. Progress bars

As with badges & medals, progress bars, as the name implies, give the user a sense of progress and accomplishment whilst completing tasks.[7][8][9]

VI. UI DESIGN ASSESSMENT BY USABILITY TESTING

Usability is a quality attribute that assesses how easy user interfaces are to use. On the Web, usability is a necessary condition for survival. If a website is difficult to use, people leave. If the homepage fails to clearly state what a company offers and what users can do on the site, people leave. The word "usability" also refers to methods for improving ease-of-use during the design process. Usability is defined by 5 quality components:

Learnability: How easy is it for users to accomplish basic tasks the first time they encounter the design?

Efficiency: Once users have learned the design, how quickly can they perform tasks?

Memorability: When users return to the design after a period of not using it, how easily can they reestablish proficiency?

Errors: How many errors do users make, how severe are these errors, and how easily can they recover from the errors?

Satisfaction: How pleasant is it to use the design?

There are many methods for studying usability, but the most basic and useful is user testing, which has 3 components: First, get holds of some representative users, such as customers for an e-commerce site or employees for an intranet (in the latter case, they should work outside your department). Second, ask the users to perform representative tasks with the design. Third, observe what the users do, where they succeed, and where they have difficulties with the user interface. Shut up and let the users do the talking.

It's important to test users individually and let them solve any problems on their own. If you help them or direct their attention to any particular part of the screen, you have contaminated the test results.[4][5]

CONCLUSION

In this paper, we reviewed four well-known user interface design rules for application systems. Both Shneiderman and Nielsen start with a rule calling for consistency in design. Both lists have a rule about preventing errors. Shneiderman rule to “permit easy reversal of actions” is about the same in Nielsen rule to “help users recognize, diagnose, and recover from errors". "make users feel they are in control" is referring the same content with “user control and freedom”. Johnson mostly focuses on designing for users needs and tasks. Gamification is a recent approach that uses in most popular software user interfaces. It is barely on increasing user engagement on software by giving them scores or playing games. The main of all of these approaches is to design an user interface such that users could easily do their jobs and engage to applications. Some of them have similar methods that the reason could be just later authors were influenced by earlier ones. Every UI design could be evaluated by usability testing in order to understand how users could easily work with systems.

REFERENCES

[1] https://en.wikipedia.org/wiki/User_interface_design

[2]hneiderman B (1999), "Jefferson's Laptop: User interfaces for universal creativity", Educom Review., May/Jun., 1999. Vol. 34, pp.34-36

[3] Scholtz J and Shneiderman B (1999), "Introduction to Special Issue on Usability Engineering", Empirical Software Engineering. Hingham, MA, USA, Mar., 1999. Vol. 4(1), pp. 5-10. Kluwer Academic Publishers.

[4] https://www.nngroup.com/

[5] M. Young, The Technical Writer's Handbook. Mill Valley, CA:University Science, 1989.

[6] Johnson, J. (2014). Designing with the mind in mind: simple guide to understanding user interface design guidelines. ISBN: 9780124079144, Morgan Kaufmann.

[7]https://tubikstudio.com/gamification-in-ux-increasing-user-engagement/

[8] K. Robson, K. Plangger, J. H. Kietzmann, I. McCarthy, and L. Pitt, "Is it all a game? Understanding the principles of gamification," Bus. Horiz.,

vol. 58, no. 4, pp. 411–420, 2015.

[9]https://www.interactiondesign.org/literature/topics/gamificationK.Elissa, "Title of paper if known," unpublished