**User-Centered Design**

Broadly speaking, User-Centered Design (UCD) is an approach to user interface design and development that focuses on the needs and tasks of the end-users of a computer system. Its purpose is to conceive a usable system, which is a system that has to be effective, efficient and enjoyable in using it.

Effectiveness concerns the “accuracy and completeness with which users achieve specified goals”; efficiency is related to how quickly the end-users perform tasks after having learnt how to use the system. Finally, enjoyment regards how much a person who interacts with the system is free from frustration and enjoys using it.

In order to design a usable system, UCD adopts a series of strategies:

• centralization of user needs as the leader of the product development

• users involvement during the UCD process

• iterative developing, which allows to built up the knowledge step by step

* involvement of many other disciplines

UCD can adopt many processes to achieve its purpose. In T-UCD, the UCD process is simplified by dividing it into three phases: user research, prototyping and evaluation. Each of them is composed by specified UCD methods that will be described in the next sections of this tutorial.

Advantages of adopting UCD to develop a system are both financial and user satisfaction. A common agreement within the industry is that UCD is considered as a fundamental factor of success for the product usefulness (Mao, 105).

Mainly, in fact in many cases there is a reduction of the development time and effort and of the support costs. That means, for example, that UCD allows detecting earlier functionalities that are useless thus avoiding to develop them: the earlier changes are detected the less costs are.

Moreover, often there is an increase of revenue and sales. That is thanks to the fact that the users of the computer system are more satisfied in using it.

The reason why there are all these advantages is that UCD allows designers and developers to have a better picture of how the system should be because there is more empathy with the users: designers and developers can put themselves in end-users’ shoes.

Concerning the main disadvantage of UCD, it is known that it may be time consuming (Mao 107). But if there are high time and budget constraints it is better to perform a bit of UCD than nothing at all.

Generally, 10% of a project budget should be invested in UCD approach, which increases in average 83% the usability of the system.(Nielsen 2008)

<http://www.usability.gov/what-and-why/benefits-of-ucd.html>

<http://www.nngroup.com/articles/usability-roi-declining-but-still-strong/>

http://www.nngroup.com/articles/return-on-investment-for-usability/

<http://www.usabilitynet.org/management/b_overview.htm>

libro 15. 20

*The state of User centered design practice* Mao

**User Research**

In order to redesigning an existing system or designing a new one that meets the needs of the end-users, it is important to understand them, their tasks and their environment where the system is or will be used. It may appear a simple task, but actually there are many aspects that should be discovered such as biography (age, family, personality, education, hobbies…), professional status (job title, work experience, current work, tasks and responsibilities, professional goals…), health (abilities and disabilities), the relation with the technology (easiness, difficulties…) and so on. At the end of this research, it should be established what aspects will be considered as the system requirements.

In the UCD process, user research focuses on these goals by applying a variety of methods such as: interview, survey, observation, persona, scenario and so on.

In the next sections of this tutorial, they will be presented by a short description, benefits and by proving best practices and some examples in order to perform them in the best way.

<http://www.usabilitynet.org/tools/mainrequirements.htm>

ch. 2 pp

**Prototyping**

During this phase, some prototypes of user interfaces are designed based on requirement previously established in the user research phase. Firstly, they will be drafted on paper (low-fidelity prototype). That allows an easy and cheap way of exploring first ideas from which will mature prototypes through iterative process of evaluation and redesign. Secondly, when the design objectives are mainly achieved a computer-based prototype (high-fidelity prototype) is created. That allows a simulation of interaction with the interface (when clicking on a button a fictive command is executed), which conveys a more realistic idea of the final computer system.

The process of designing interfaces should be based on design guidelines. The most common are listed and summarized here below (Preece):

**Visibility** – The more visible functions are, the more likely users will be able to know what to do next. Incontrast, when functions are "out of sight," it makes them more difficult to find and know how to use.

**Feedback** – Feedback is about sending back information about what action has been done and what has been accomplished, allowing the person to continue with the activity. Various kinds of feedback are available for interaction design-audio, tactile, verbal, and combinations of these.

**Constraints** – The design concept of constraining refers to determining ways of restricting the kind of user interaction that can take place at a given moment. There are various ways this can be achieved.

**Mapping** – This refers to the relationship between controls and their effects in the world. Nearly all artifacts need some kind of mapping between controls and effects, whether it is a flashlight, car, power plant, or cockpit. An example of a good mapping between control and effect is the up and down arrows used to represent the up and down movement of the cursor, respectively, on a computer keyboard.

**Consistency** – This refers to designing interfaces to have similar operations and use similar elements for achieving similar tasks. In particular, a consistent interface is one that follows rules, such as using the same operation to select all objects. For example, a consistent operation is using the same input action to highlight any graphical object at the interface, such as always clicking the left mouse button. Inconsistent interfaces, on the other hand, allow exceptions to a rule.

**Affordance** – is a term used to refer to an attribute of an object that allows people to know how to use it. For example, a mouse button invites pushing (in so doing acting clicking) by the way it is physically constrained in its plastic shell. At a very simple level, to afford means "to give a clue" (Norman, The Design of Everyday things, Basic Books, 2002). When the affordances of a physical object are perceptually obvious it is easy to know how to interact with it.

Producing prototypes of the target system allows making changes before it will actually be implemented. Changes accomplished after the system is coded have a higher cost in terms of time and money.

Involving typical end-users in this phase is really important because designers are not the real end-users and may design prototypes that do not meet their needs.

At the end of this phase, few designs should be chosen in order to use it for the next phase: evaluation.

The methods to accomplish these tasks are for example sketch and storyboard that are included in low-fidelity prototype group and other methods under high-fidelity prototype. They will be described in the next section of this tutorial by a short description, benefits and by proving best practices and some examples in order to perform them in the best way.

Source: Preece, J., Rogers, Y., Sharp, H. (2011), Interaction Design: Beyond Human-Computer Interaction, New York: Wiley & Sons, p.26-29

<http://www.nngroup.com/articles/paper-prototyping/>

<http://www.usability.gov/how-to-and-tools/methods/prototyping.html>

<http://www.usabilitynet.org/tools/design.htm>

libro ch.3

**Evaluation**

Finally, the third phase consists in evaluating if the prototype(s), which has been established in the previous phase, is/are usable: effective, efficient and satisfactory in using it. The evaluation more commonly performed is by involving the end-users. Typically, they are supposed to accomplish some tasks with the prototype, to give feedback about problems and to explain their expectations. This phase is useful to designers in terms of understanding of what should be improved.

Furthermore, it is also possible to hire usability experts that check if the prototype achieves the usability heuristics.

At the end of this phase, designers should have gathered data in several formats such as comments, notes and audio or video recordings.

Some methods performed in this phase are usability evaluation, heuristic evaluation and accessibility evaluation.

They will be described in the next section of this tutorial by a short description, benefits and by proving best practices and some examples in order to perform them in the best way.

<http://www.usabilitynet.org/tools/expertheuristic.htm>

http://www.usabilitynet.org/tools/evaluate.htm

ch. 4