OMONIYI FOLUKE TEMITOPE

16/SCI01/035

DEPARTMENT OF COMPUTER SCIENCE

COLLEGE OF SCIENCE

CSC406

**ASSIGNMENT**

**DISCUSS THE GOALS OF HCI**

The term Human Computer Interaction (HCI) was adopted in the mid-1980s as a means of describing this new field of study. This term acknowledged that the focus of interest was broader than just the design of the interface and was concerned with all those aspects that relate to the interaction between users and computers. The goals of HCI are to produce usable and safe systems, as well as functional systems. These goals can be summarized as ‘to develop or improve the safety, utility, effectiveness, efficiency and usability of systems that include computers’ (Interacting with computers, 1989). In this context the term ‘system’ derives from systems theory and it refers not just to the hardware and software but to the entire environment---be it organization of people at work at, home or engaged in leisure pursuits---that uses or is affected by the computer technology in question. Utility refers to the functionality of a system or, in other words, the things it can do. Improving effectiveness and efficiency are self-evident and ubiquitous objectives. The promotion of safety in relation to

computer systems is of paramount importance in the design of safety-critical systems.

Some of these goals include;

**Efficiency** It refers to the way a system supports users in carrying out their tasks. Safety It involves protecting the users from dangerous conditions and undesirable situations. In relation to the first ergonomics aspect, it refers to the external conditions where people work. For example, where there are hazardous conditions---like x-rays machines or chemical plants---operators should be able to interact with and control computer-based system remotely. The second aspect refers to helping any kind of user in any kind of situation avoid the danger of carrying out unwanted action accidentally. It also refers to the perceived fears users might have of the consequences of making errors and how this effects their behavior to make computer-based system safer in this sense involves: **.** Preventing the user from making serious error by reducing the risk of wrong keys/buttons being mistakenly activated (an example is not placing the quit or delete-file command right next to the save command on a menu.) and **.** Providing users with various means of recovery should they make errors. Save interactive systems should engender confidence and allow the users the opportunity to explore the interface to carry out new operations.

**Utility** It refers to the extent to which the system provides the right kind of functionality so that user can do what they need or want to do. An example of a system with high utility is an accounting software package providing a powerful computational tool that accountants can use to work out tax returns. An example of a system

with low utility is a software drawing tool that does not allow users to draw free hand but forces them to use a mouse to create their drawings, using only polygon shapes.

**Learnability** It refers to how easy a system is to learn to use. It is well known that people do not like spending a long time learning how to use a system. They want to get started straight away and become competent at caring out tasks without to much effort. This is especially so far interactive products intended for everyday use (for example interactive TV, email) and those used only infrequently (for example, video conferencing) to certain extent, people are prepared to spend longer learning more complex system that provide a wider range of functionality (for example web authoring tools, word processors) in these situations, CD ROM and online tutorials can help by providing interactive step by step material with hands-on exercises. However many people find these tedious and often difficult to relate to the tasks they want to accomplish. A key concern is determining how much time users are prepared to spend learning a system. There seems little point in developing a range of functionality if the majority of users are unable or not prepared to spend time learning how to use it.

**Memorability** It refers to how easy a system is to remember how to use, once learned. This is especially important for interactive systems that are used infrequently. If users haven’t used a system or an operation for a few months or longer, they should be able to remember or at least rapidly be reminded how to use it. Users shouldn’t have to keep relearning how to carry out tasks. Unfortunately, this tends to happen when the operation required to be learning are obscure, illogical, or poorly sequenced. Users need to be helped to remember how to do tasks. There are many ways of designing the interaction to support this. For example, users can be helped to remember the sequence of operations at different stages of a task through meaningful icons, command names, and menu options. Also,

structuring options and icons so they are placed in relevant categories of options (for example, placing all the drawing tools in the same place on the screen) can help the user remember where to look to find a particular tool at a given stage of a task. “Don’t Make me THINK, is the key to a usable product”