

# [Web design](https://assignbuster.com/web-design/)

[](https://assignbuster.com/)[Design](https://assignbuster.com/essay-subjects/design/)

UCD stands for user centred design and forms a basis of how web design is approached. Similar to Rapid Application Development (RAD), it is just another form of design methodology. It focuses mainly on how users interact with the front end of the website design, and in turn leading to a much more usable, and easy to learn website. UCD involves users in every phase of the product development cycle, instead of waiting for evaluation at the finished product, UCD provides user feedback throughout the development of the system.

UCD is not a new methodology, it was originally touted by Apple in a video in 1989 called " Apple World Builder". Even IBM and Microsoft had picked up on its potential and created many websites devoted to it. Within UCD, the focus is shifted from new technology to users, and from user interfaces, to actual users. The focus revolves around understanding and building around the users and their usage. We must not forget however, that UCD is not something that covers every single aspect of web design, the strategies and marketing are obviously not included, but without good user interaction which is aiding in the completion of the tasks, one could say that the product would struggle to succeed in the marketplace in the first place.

There are many benefits of UCD, an article published by Susan Dray in the January 1995 edition of " interactions", listed a group of benefits which she thought would come as a result of a UCD;   
UCD is not a process or methodology that can be easily defined and suited to every case, different organisations have different forms of it. The common basis of UCD however, remains central for any different type or version.

These include early focuses on users and the tasks, experimental measurements of the usage, and an iterative design where the product is designed, and evaluated. In this course, UCD is represented as a five stage cycle similar to the above, this consists of; 1. Evaluating the usability2. Conducting a HTA 3. Establishing requirements 4. Prototyping through a storyboard 5. Evaluating through user observation Body of the Report The aim of this essay is to demonstrate a better and deeper understanding of the UCD process, by stepping through each of the five steps above and covering alternative points as well as understanding each of the five issues from a better perspective.

Jakob Nielsen's heuristics was a topic which featured heavily in exercise 1. A heuristic evaluation of an online diary system was requested. These heuristics are essentially a set of recognised usability principles. Through using these guidelines a list of good and bad points can be reviewed about any particular website. The heuristic evaluation centered along the 10 key principles, and below is what I have finalised about what I think Nielsen meant by each of them. I have tried to conduct a better understanding of each heuristic, compared to what I thought originally in week three, as well as a few checkpoints for each heuristic.

1. Visibility Of System Status - This principle basically revolved around the notion of letting the user know where he stood exactly in the system, so the user knew what page he was on, what was happening, and where to report feedback to. Evaluating a site according to this principle could involve checking to see if some of the following points are met; are selected icons visible compared to unselected icons?, does each page contain a separate header describing the contents of that particular page?, and does the menu appear on the same place in each page?.

2. Match Between System And The Real World - This heuristic referred to the language of the site. Explaining that it should be easy for the user to understand, containing words and phrases that the user is familiar with. Evaluation checkpoints for this include; are the icons familiar?, or do the menu choices have a natural sequence of order? 3. User Control And Freedom - This heuristic asked for the user to be provided with more control, allowing them to be free to select and sequence tasks rather than always having the system to do it for them. This principle also indicated in providing an option for the user to leave a particular situation on the site via an 'emergency exit'. Certain points to check for in the site could involve; when a user has completed a task does the system wait for a signal before processing it?

4. Consistency And Standards - This principle basically asked that the system should make sure that the same words and actions meant the same thing, in all situations. Making sure that your system followed this guideline correctly involved looking out for features such as; are all of the icons labeled?, and are there too many icon types? 5. Help Users Recognise, Diagnose And Recover From Errors - This principle basically asked for the system to provide a good form of error messaging when a problem did arise, and that the messages should be expressed in plain English instead of complicating codes, as well as providing the user with an explanation of how to solve the problem. A checklist for this would involve the following; are prompts too brief and unexplanatory?, when there is an error, is sound used?, If comical error messages are used are they inoffensive and appropriate?.

6. Error Prevention - This heuristic follows the notion that a good design should not have any problems in the first place. 'Error Prevention' deals with preventing errors from occurring. In order to follow this heuristic successfully you would have to check to see for certain points such as; are menu choices distinctive and logical?, and, will navigation between windows be simple and clear if the system is displaying multiple windows?.

7. Recognition Rather Than Recall - This heuristic involved making actions, objects and options visible. The user should not have to remember information from one part of the system to another part. The user should be able to 'recognise' rather than have to 'recall'. For example, a user should be able to recognise certain icons through the system instead of trying to recall what they are. Other check points include; making sure that items such as messages and prompts are placed exactly where the eyes are likely to be looking on the screen. Also does the data display start and appear in the top left hand corner?.

8. Flexibility and minimalist design - This heuristic talks about providing accelerators to speed up the interaction for expert users. The system should still however, cater for both inexperienced and experienced users. This principle means to tailor or speed up frequent actions used by the expert users. This could be in the form of shortcut keys. In order to pass this heuristic, the system could be checked for some of the following points; As the system is catered for both types of users, are different levels and types of error messages available and are they detailed sufficiently for each?. Also can users make up their own synonyms for commands?.

9. Aesthetic and minimalist design - This heuristic was key to Nielsens whole theory of simplicity. It revolves around the notion of the system containing only necessary information and not unnecessary information which is irrelevant, and in turn diminishing the visibility of the information which is relevant.. Checklists for this heuristic would involve checking to see if all icons are set visually distinct, and whether the icons are standing out from the background. Also is only the necessary information essential to the decision making, being displayed on the screen, and is ALL the necessary information displayed.

10. Help And Documentation - It can be necessary to provide help and documentation in all systems. However it is obviously better if the system is ready for use without the use of any type of documentation. When help information is provided it should be in the form of concrete steps, not be confusing, or contain too much information. It should also be easy to search and follow, whilst remaining focused on the users task. Checkpoints to see if your system compiles with this heuristic could be as follows; do the help instructions follow the actual sequence of the users actions?, also do data entry boxes come with completion instructions, or is it confusing?.

Jakob Nielsen's main design principle revolved around simplicity, he recommended removing elements of the design which were unnecessary or elements that did not affect the actual functioning of the website. In argument against this case, a certain level of graphics and icons can also make it pleasing to the users eye by being more attractive than the usual boring and standard website, hence, the key to this point being to find just the right balance between the two. A key feature that I noticed about his heuristic evaluation was that different people noticed different usability problems and if I was to repeat the heuristic evaluation of a website, I would involve other people to aid in evaluating the website in order to improve the overall effectiveness. Exercise two involved conducting a hierarchical task analysis of the manual task of keeping a diary.

In hierarchical analysis, an instructional designer decomposes and breaks down a task from its top to its bottom, and amongst these tasks a hierarchical relationship is shown. So basically, if someone wanted to perform task six at the bottom of the relationship, they could not perform it until they had completed tasks 1-5 first. Creating a HTA involves grouping tasks together, labeling them according to the functions they complete and then decomposing each task into subtasks, asking yourself along the way " what would the user have to do in order to achieve this task?".

Following out a HTA involves three basic steps; starting an analysis, progressing it, and finalising the analysis. Through starting the analysis you specify the actual main task, break that down into subtasks, and then plan how the subtasks are connected. Progress of the analysis involves deciding how much detail should be involved, and whether or not each task should be continued or not. When finalising the analysis, you check to see if all alternatives are covered, and all the subtasks are linked directly.

Evaluating a HTA includes checking for an adequate number of tasks and checking for an appropriate depth of levels. Also checking to see if the validity and accuracy is good, for example; does the analysis relate well to the actual learning process?. Checking for consistency in grouping tasks on the same level in the hierarchy is another evaluation point. HTA creates a graphical representation of the decomposition of tasks into subtasks, in the process setting a hierarchy of actions and plans.

HTA has many advantages which include allowing the design of supporting systems to be able to offer alternative methods or ways of performing certain parts of an individual task. Also if subtasks are decomposed into a more detailed architecture then this could enable a very precise and accurate performance analysis. HTA helps in understanding user requirements regarding the iterative design and the allocation of duties.