Heuristic evaluation

Introduction

Any software cannot be possibly released into the market unless it has been tested and validated well enough. Usability testing is one such test that the system should ideally pass through before being deemed fit for use. As more and more features are being packed into a system, the resultant complexity sometimes tends to overlook the ease of use of the application. This is where the importance of usability testing comes in all the more as any system can't be considered good unless it is efficient and easy to use. There are many ways of carrying out usability testing, one such method is **Heuristic evaluation** , which is a discount usability inspection method for computer software that helps to identify usability problems in the user interface design. Based on a known set of criteria, a group of usability experts will go through and evaluate the system. As a result there will be a list of observations (in order of severity). Because the evaluation is made by experts who are familiar with all usability issues, the results don't concentrate on the properties of the interaction between an actual user and the product, but the interface functionality.

Possible list of heuristics

Visibility of system status

It concentrates on how well the status message and information regarding the system are displayed to the user. It also looks at possible ways to improve the notification mechanism from usability perceptive.

Match between system and the real world

Any system can't exist in vacuum so to speak, it has to relate well to the real world where it is supposed to be used. This is where some of the domain expertise comes into picture where the real life scenarios have to be taken care of properly in the system. For instance banking software should consider the possible scenarios when the user is trying to withdraw cash, insufficient funds, maximum transaction amount limit etc.

User control and freedom

How much freedom of decision making does the system give to the user and how much of the decision making is inbuilt into the system is another aspect which guides usability matrix. There are certain systems where it makes lot of sense for the system to be doing the decision making for the user while in some the user wants to take control. Games, media player etc could be examples of the latter category.

Consistency and standards

The system has to comply with the conventions like style guide etc and has to be consistent. For instance you can't possibly decide to show an error message in one case while simple gloss over the error in another case. An also commonly accepted convention like soft key placements etc needs to be adhered to so the user doesn't end up being confused.

Error prevention

There are some cases where running into an error case is unavoidable like network failure etc while there are some which can be prevented if the logical checks etc have been carried out well. No user likes the idea of using a system which seems to throw up error messages for almost every single operation. It is important to minimize errors where they are possible to make the user feel comfortable using the system.

Recognition rather than recall

Recognition of possible error conditions, rather than getting into a fire fighting mode after an error has occurred is always a good idea from a usability perspective.

Flexibility and efficiency of use

Any software well designed and developed should be efficient and easy to use. The user should always feel that the software would do the intended services well to his expectations, and there should be as less surprise from a user's point of view as possible.

Aesthetic and minimalist design

Aesthetics play a crucial role in deciding the overall usability of a product. A system with nice look and feel will always score over

a similar product with not so well done ui. It is very important to sleek up the interfaces as they say ‘looks can be deceptive’ and sometimes first impressions become the last impression. Another thing to take into account is that the interface should be minimal and not verbose; too many fields and details will only confuse the user.

Help users recognize, diagnose, and recover from errors

This check looks at whether the users are provided enough help inbuilt to understand the errors and whether they are able to solve the errors on their own taking the help of the documentation provided with the software.

Help and documentation

Every system should come bundled with help and proper documentation to assist the user in using the features of the system. The help should be worded in user’s parlance so that they can understand what is being talked about. For details about help manuals check Things to remember when writing Help text or Manuals

Advantages

A heuristic analysis is a cheap and less time taking exercise to conduct, and can be done by even one expert who is well versed with the key components of the system and understands the user’s domain. It can be carried out iteratively thereby helping in diagnosing any variance between the expected and the actual so that they can be rectified at an earlier stage in the development process.

Criticism

Since the heuristic evaluation is undertaken by the expert, the efficacy of the exercise also depends to a large extent on the knowledge and competence of the expert carrying out the evaluation. Sometimes it is considered a "one sided" review as there are no additional view points that are taken into considetation. There are chances that something might appeal to the expert while might not appeal to the others, but since the expert agrees/disagrees to the content/user interface they are ok'ed during the heuristic evaluation.

Additional Resources

Useit.com

usability net

Wikipedia on heuristic evaluation

step by step guide to heuristic evaluation

--- Edited by Mayank on 21/06/2009 ---