“Usability Testing - Starting With Satisfied Customers”

Chris Ambler

Presentation

T24

International Conference On
Software Testing, Analysis & Review
November 19 - 23 Stockholm, Sweden

Thursday 22nd November, 2001

Thursday 22 November 2001
T24

“Usability Testing – Starting With Satisfied Customers”

Chris Ambler

Hailing from West Yorkshire, Chris has been involved in the computer industry since 1976. After twelve years in Operations Management, working on ‘big booming mainframes’, he took a study break for three years and obtained an honours degree in Computer Science from The University of Bradford as a mature student. His initial involvement in testing was in the defence sector testing software for aircraft and missile systems. Back in the commercial environment, there followed many years as a contract test manager, working on projects in the financial, government and services sectors. Moving back into permanent employment, He then set up and ran a global testing practice for a major American solutions company with testing assets and resources worldwide. He then joined SIM Group as a Business Unit Manager, currently managing major web testing and application testing projects in the finance and banking arena. Chris is married to Carol, has three teenage children, a Westie called Donald and now lives on the Sussex coast.

Usability Testing - Starting with Satisfied Customers

Chris Ambler

Chris.Ambler@Simgroup.com

2 ‘true’ statements

- Ignoring customer requirements has never been a good way to maximise your revenue

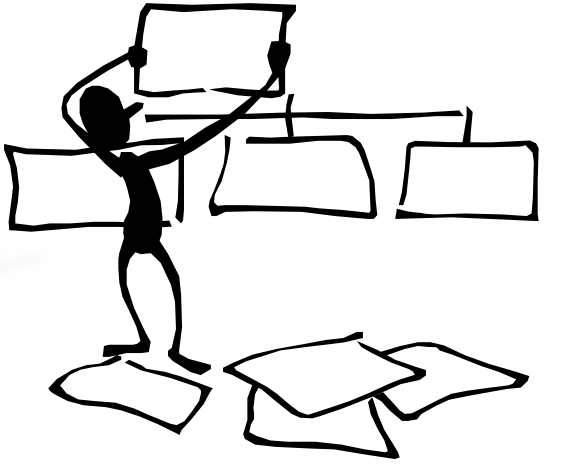
- UAT traditionally fails the business anyway – why do we really need ‘operational procedures’ and ‘training’ to get round system shortcomings?

What we will discuss….

- Why are Customers ‘different’ to Users
- How to identify Usability Requirements
- Product gains from this new element of testing

What is a Customer?

- Customers define the requirements
- Customers pay the bill !!
- Customers have specific drivers
  - Operational
  - Financial



What do Customers want?

- Usability-systems that are easy to learn and remember, systems that enhance our productivity, and systems that are error resistant and friendly.
- Interoperability-systems that are able to interact with other systems and exchange data with predictable results.
- Scalability-information technology is changing so fast, we want our capabilities to evolve with the changes, without the need to replace our systems with each new development.
- Reliability- systems that perform as expected.

What is a User?

- Users are quite often our customers ‘customers’
- Users will ‘make or break’ the products on offer
- Users can be very ‘fickle’
- Users ‘use’ the application

What is a User?

- Once you've lost a user, you've lost him for good. The costs of flipping to another Web site are so low it doesn't make any sense for people to go back to a site that failed them the first time.

How to identify Usability Requirements

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

Even better than good error messages is a careful design, which prevents a problem from occurring in the first place.

Make objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

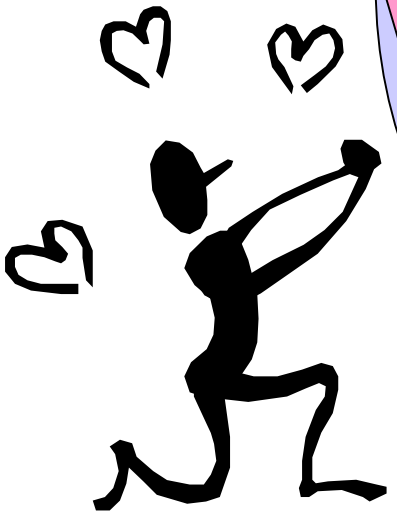
Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

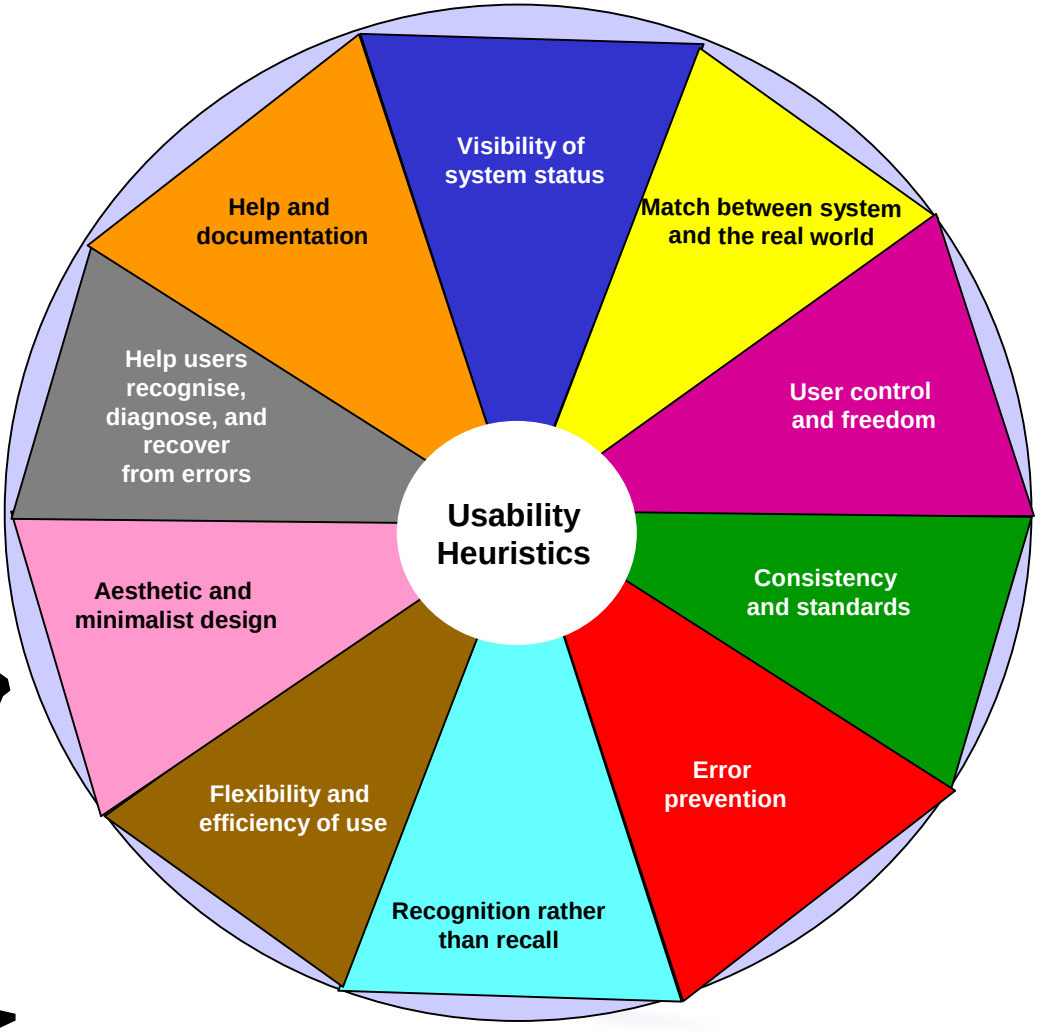
Dialogues should not contain information, which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

How to identify Usability Requirements





Some famous examples

- NASA’s 1969 Apollo 11 moon landing ‘Error 101’
- Windows 98 asks users to press "start" to stop.
- On the Apple PowerBook, the "on" switch doesn't say "on," it just has a little arrow pointing left. I hope the fire control systems on our jet fighters don't have a little arrow pointing left, but rather say something like "push here to blow the other guy up."
- On Windows NT, you press control, alt, delete to start using the system. With Windows 3.1 and MS-DOS, you pressed the same three keys to reboot.
- People are still scanning their key boards looking for the "any" key.

Product gains from this new element of testing

- Increased trust
- Increased customer satisfaction
- Increased revenue !!

Watch and understand what users *actually* do. Do not believe what users *say* they do. Definitely don't believe what users predict they *may* do in the future.

“People don't want to compute. They want to write reports, find information, create art, prepare budgets, manage their schedules, and send messages to one another.

Users are not stupid
So Watch and Learn !!

Usability Testing - Starting with Satisfied Customers

Chris Ambler

Chris.Ambler@Simgroup.com