User Experience/Universal Design

User centred design

This is part of the Design Stage of the Software Development cycle.

User centred design prioritises the needs of the user.

A design team should always consider the user and involve them in the process.

A design team will work with a list of user requirements and always be considering

- who the users of the software will be
- how to make the software optimal for all

Two considerations for user centred design are **user experience (UX) and universal design (UD)**

User Experience (UX) Design

UX Design is the process of creating products that provide meaningful and relevant experiences to users. This involves the design of the entire process of acquiring and integrating the product, including aspects of branding, design, usability and function.

(www.Interaction-Design.org)

UX Design

Products that provide great user experience (eg iPhone) are designed with not only the products consumption or use in mind but also the entire process of acquiring, owning and even troubleshooting it.

Similarly, UX designers don’t just focus on creating products that are usable, they concentrate on other aspects of the user experience such as pleasure, emotion, efficiency and fun before during and after use.

Consequently, there is no single definition of a good user experience. Instead, a good user experience is one that meets a particular user’s needs in the context where they use the product.

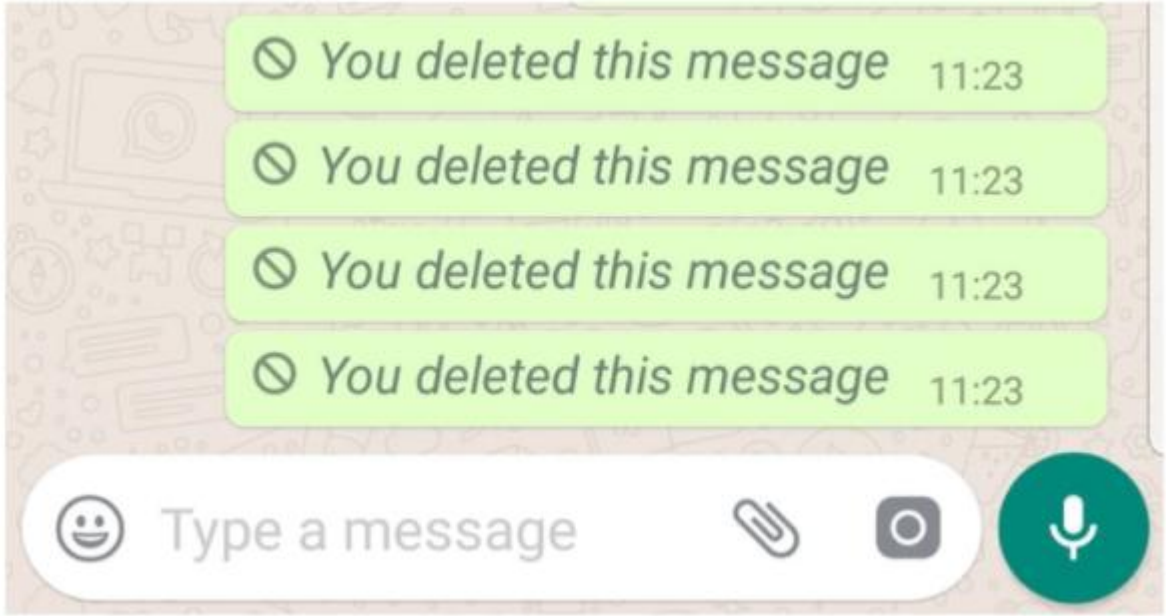
"No product is an island. A product is more than the product. It is a cohesive, integrated set of experiences. Think through all of the stages of a product or service – from initial intentions through final reflections, from first usage to help, service, and maintenance. Make them all work together seamlessly."

- Don Norman, inventor of the term "User Experience"

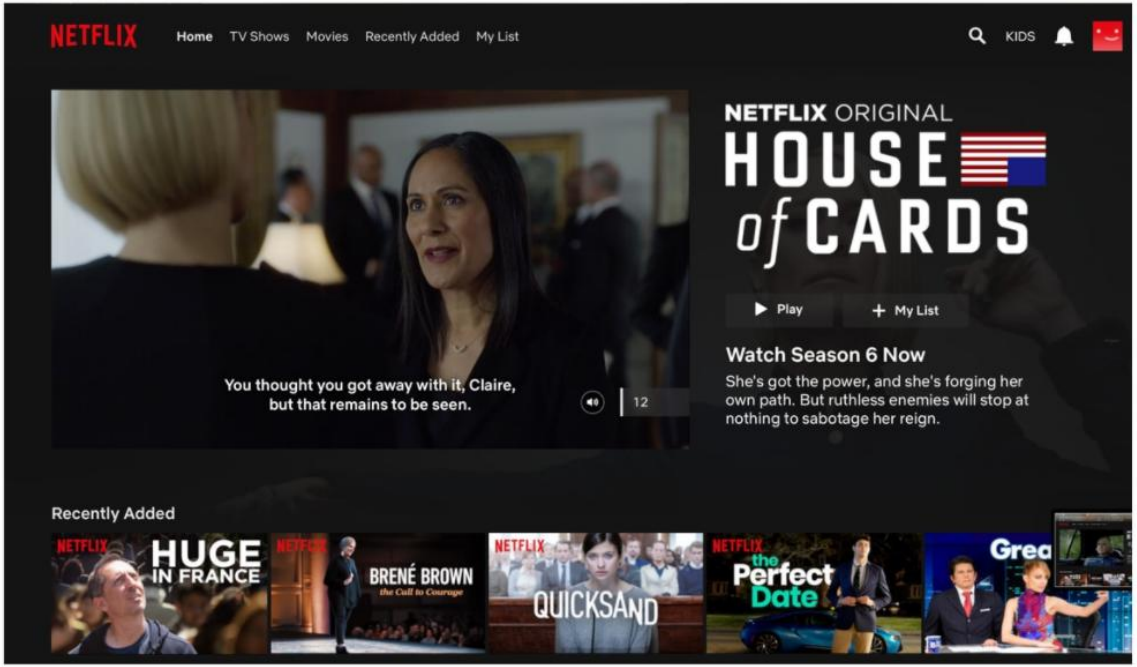
The Why, What and How of UX Design

Even the best get it wrong!

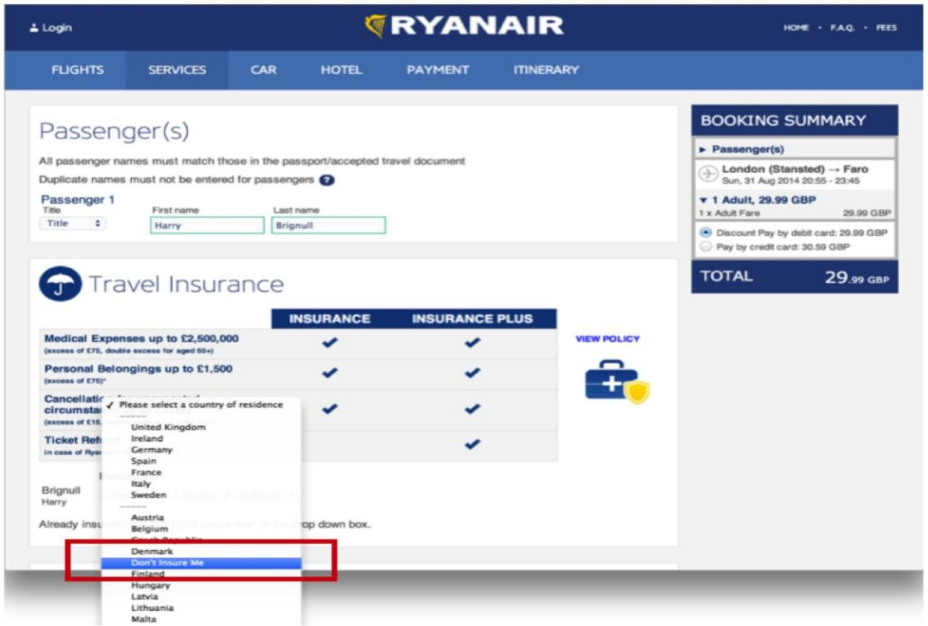
Whatsapp delete message feature



Netflix hover auto play



Ryanair's booking platform



Apple’s storage management

User Interface (UI)

For software, the User Interface (UI) is normally a graphical interface.

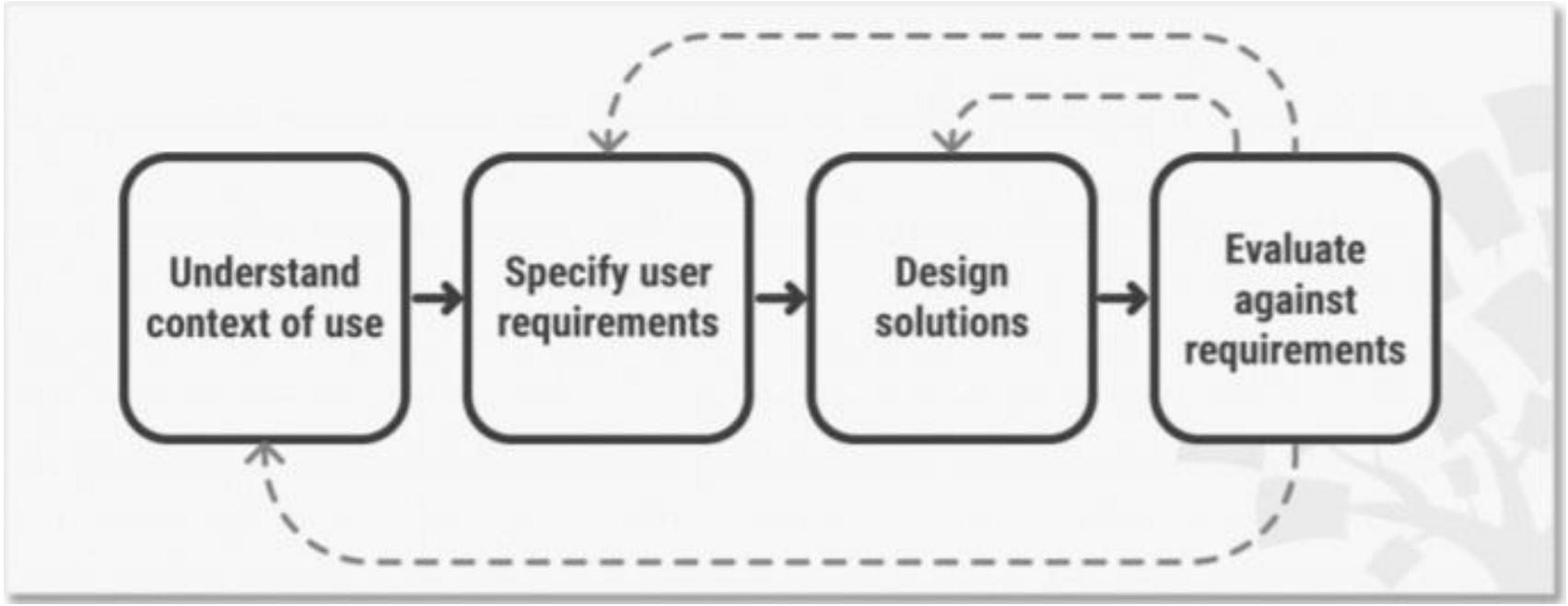
This can be the Operating System on your device or a website.

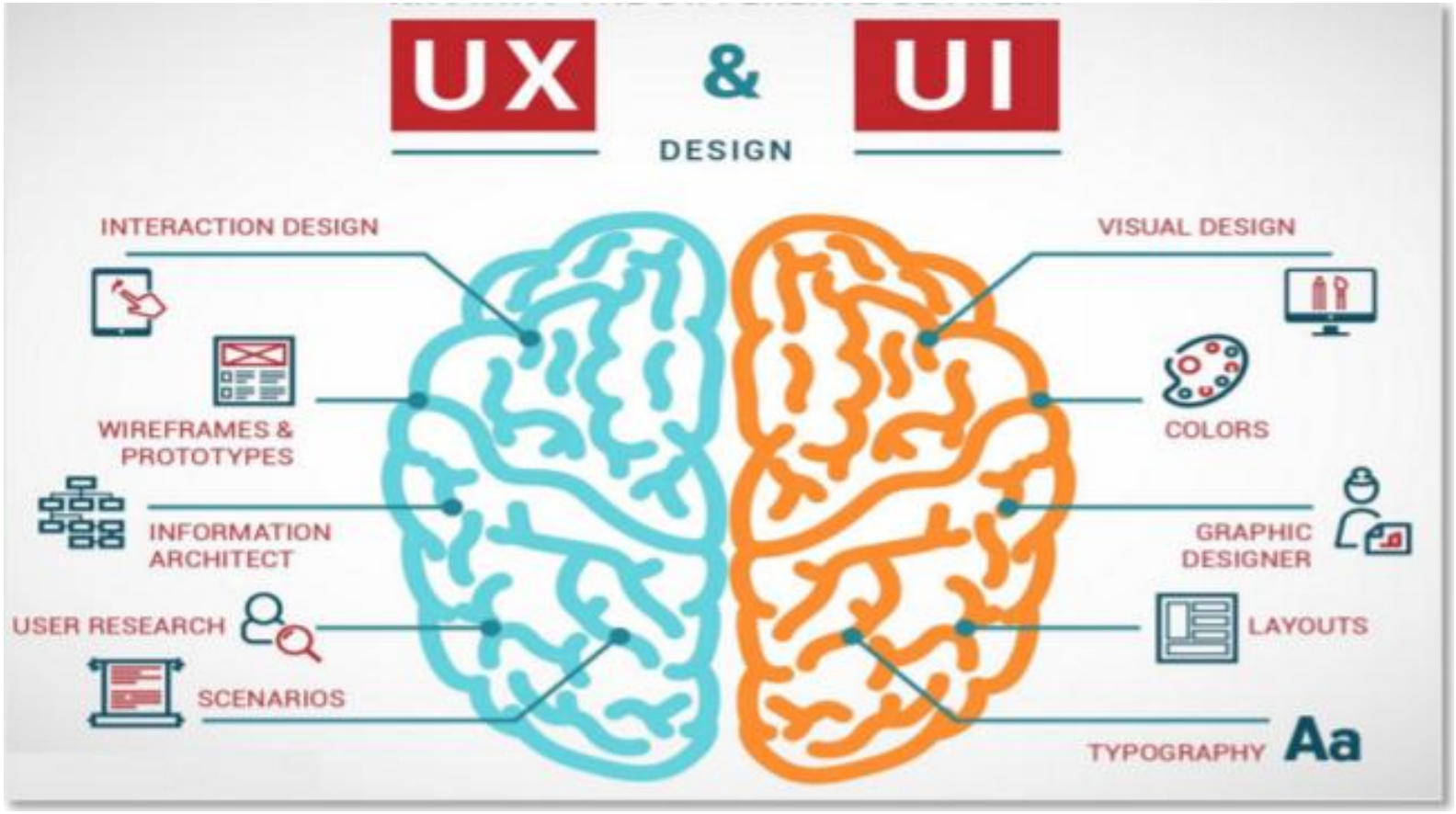
UX is the users experience of using the product.

The UI is the interface through which that experience takes place.

User Interfaces are necessary for people to use computers. They can take different forms depending on the design decisions involved.

The design team need to carefully consider their users, their users requirements and the requirements of the client in order to make the right design decisions.

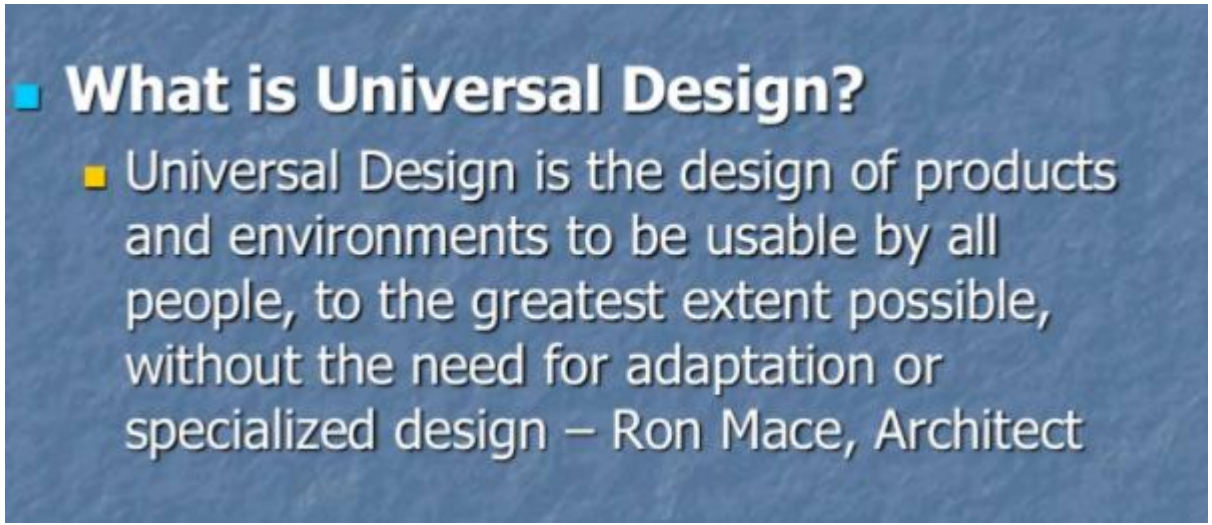

Understand context of use
Specify user requirements
Design solutions
Evaluate against requirements


UX
&
UI
DESIGN
INTERACTION DESIGN
WIREFRAMES &
PROTOTYPES
INFORMATION
ARCHITECT
USER RESEARCH
SCENARIOS
VISUAL DESIGN
COLORS
GRAPHIC
DESIGNER
LAYOUTS
TYPOGRAPHY
Aa

Universal Design

Universal design is the process of designing a product so that everyone can understand, access and use it, regardless of their age, size and ability.

This may not always be achievable but it is a worthy goal.



Universal Design Principles

Universal design has seven generally agreed principles that were developed in 1997 by a group of architects, product designers, engineers and environmental design researchers.

Most of these principles apply to most software, web pages etc.

The seven principles of Universal Design can be found here:

http://universaldesign.ie/What-is-Universal-Design/The-7-Principles/

Example of Universal Design

Quorum Programming Language - https://quorumlanguage.com/ is an evidence based programming language which means that the decisions made in designing and refining the language are based on evidence.

The following slides will go through the 7 Principles of Universal Design and how they are applied to Quorum.

1. **Equitable Use**

Quorum is suited for screen readers because the syntax of the language is designed to be simple and doesn't contain unnecessary words or punctuation. It is used by many blind or visually impaired programmers.

1. **Flexibility**

It is available through a web interface and a stand alone IDE. It has a visual editor that is designed to be used by many with disabilities.

1. **Simple and intuitive use**

The syntax was designed to be simple which means that the language is better for screen readers, captioning and for those with and without learning disabilities. Example code for a `for` loop in Quorum is `repeat 5 times`

4. Perceptible Information

Quorum interfaces with the operating system to give very specific information about what is happening on the screen. This allows the operating system to feed this information to other devices and programs eg screen readers. If Quorum didn’t provide this information to the operating system, such devices wouldn’t work.

5. Tolerance for Error

Quorums error messages are designed to provide clear, actionable information when errors occur.

6. Low physical effort

Simple syntax and clear error messages mean less effort is required to read, speak and listen to commands and errors. There is also a magnify feature to help the visually impaired.

7. Size and space for approach and use

Quorum has shortcut keys for users with mobility impairments - for example if a user only has the use of one hand, the layout of the keyboard can be remapped.

Adaptive technology

Adaptive technologies provide features or feature enhancements to existing technologies to aid their use for people with impairments and disabilities.

User centred design and universal design are very closely related to the concept of adaptive technology.

Examples of adaptive technology are screen readers, alternative keyboards, text to speech tools.

Assistive technology

Assistive technology is similar to adaptive technology. It refers to something that can help people with disabilities - to assist in using the technology.

An example is a Magnifier app which allows users to magnify anything that is on their screen.

Designing for accessibility

It can be challenging to design for accessibility but it is important to realise that making software accessible for one group can make it better for other groups too.

Examples:

Captioning provides text description of a video soundtrack including background noises and dialogue. These were originally designed for people with a hearing impairment. This data can also be inputted to screen readers to help those that are visually impaired.

People with learning disabilities may also use screen readers.

Extra Reading: Case Study

Pol Kuijken’s case study on his “Google Design Exercise: Solving the shelter” problem when he was applying for a position at Google.

https://medium.com/@polkuijken/pet-adoption-8798b14af117

Extra Reading: UX Process

Robert Dumitru

Don Norman

Jakob Nielsen

Ben Shneiderman