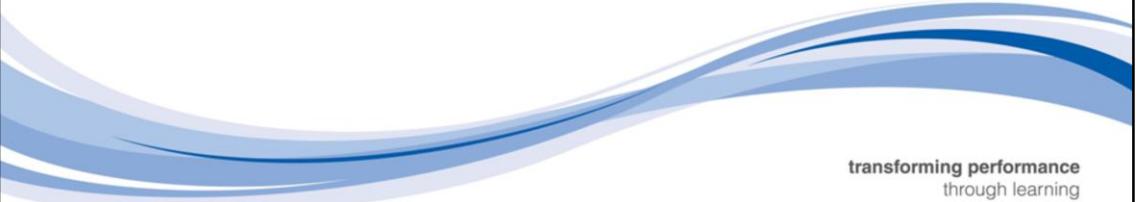




UX Fundamentals

Introducing User Experience



A decorative graphic at the bottom of the page features several thin, flowing blue and white wavy lines that curve from left to right across the page.

transforming performance
through learning

Introduction

- **The UX Problem**
 - The Better Horse
 - The Dancing Bear
- **Defining UX**
 - What is UX
 - What is Usability
 - What is UCD
- **Starting to think about methodologies**
 - Contemplating how to design for UCD

The Better Horse Argument

- Consider the following paraphrased quotation:

If I had asked my customers what they wanted,
they would have asked for a faster horse

- This is usually brought up by people claiming that
 - "users don't know what they want"

- Things to consider:

- It is true that most users do not actually know what they want
- Your job is not to ask what they want, but deliver what they need
- Henry Ford did give them a better horse

3

"If I had asked my customers what they wanted, they would have asked for a faster horse"

Henry Ford never said this.

This is usually brought up by people claiming that "users don't know what they want".

Things to consider:

It is true that most users do not actually know what they want because they do not know what is possible.

Your job is not to ask what they want, but deliver what they need.

Part of determining what they need may involve asking them what they want.

Henry Ford gave them a better horse.

It was called the Model T

The Dancing Bear

- **Failing in UX can lead you to create a Dancing Bear product**
 - Where a great idea triumphs over poor design
 - A product so good users can not do without it
 - Even if it has terrible interaction design



4

We are moving from a world where UX and what the user wants have tried to be considered but often incorrectly. There are a role call of products that have great ideas but poor usability. The process for using them, for the user using them – has been misunderstood. You have what is called a dancing bear product – a product that users can not do without but does not do it very well.

What is User Experience?

"I invented the term because I thought Human Interface and usability were too narrow: I wanted to cover all aspects of the person's experience with a system, including industrial design, graphics, the interface, the physical interaction, and the manual."

– Don Norman

"User experience" encompasses all aspects of the end-user's interaction with the company, its services, and its products.

– Don Norman & Jakob Nielsen

Make things easy!

What is User Experience?

- **User Experience is the emotional response to using a:**
 - Product
 - System
 - Service
- **It includes the user's perceptions of**
 - Utility
 - Ease of use
 - Efficiency
- **It is subjective, due to the nature of perception,**
 - It is also dynamic and constantly evolving due to new innovations
 - Changing the context through which users perceive the product

Why User Experience Matters

- **The user is the make or break for products and processes**
 - The old mantra of enterprise technology ‘build it and they will come’
 - The users of today’s technology challenge this mantra
 - They are accustomed to experiences using sites that are
 - Relevant, personalized and easy to use
- **If they don’t get these things from you they will find alternatives**
 - There are always other options
 - Being “best” is subjective
 - A user who has a good experience is less likely to look for alternatives
- **Users consider a better experience to be superior**
 - Even to a more technically adept product (within reason)

7

There are always other options

E.g. MS Office is not the only Office Suite available

It is rare to have a truly unique product. Whatever service you provide will not be without competitors for long in a viable market; The only truly unique thing that you can rely on is the experience that you craft.

Being “best” is subjective

“Best” depends on what the user needs, what they like, and what they are used to/already doing.

Compare major operating systems; MacOS, Windows and Linux on the desktop; iOS, Windows Mobile and Android on Mobile. Their broad capabilities are for the most part fairly similar to their counterparts, but they have crafted very different experiences to each other. Which is best? It depends what you are looking for.

A user who has a good experience is less likely to look for alternatives

How many basic office users still purchase MS Office for the Mac despite there being Mac alternatives? How many of them pay for Office Suites even though there are free alternatives easily available?

Why do people switch from one to another? Usually because they have a need that is not being met.

What Are the Foundations of User Experience

- **Good design makes peoples lives better UX sets out to achieve this**
 - We aim to improve the lives of your users in some way,
 - Otherwise you are doing it wrong
- **The foundation upon which everything is built is Usability**



Consider the worst Usability you have encountered on the web?

8

Activity - Think of a poor experience you have had and why?

Usability

- **Usability determines the extent to which a product satisfies:**
 - A user's ability to achieve a stated goal
 - In an effective and efficient manner
- **Every User Experience Designer needs to focus on Usability,**
 - Poor usability will undermine everything else about the design
 - it is not possible to design a good experience if it frustrates the user
- **To understand Usability you will need to think about**
 - Cognitive Limitations
 - Content Considerations
 - User Psychology

9

Cognitive Limitations: Memory limits (we only reliably remember 4 things), Inattention blindness, we do not multitask (no matter what people think), ageing slows the brain processing speed but gets better at accuracy, users scan and interpret based on expectations, memories are highly inaccurate. People have to use information to make it stick. It is easier to recognise information than the remember it. Memory is resource heavy. We rebuild our memories every time we recall them.

Content considerations: Imagine if the tools for writing text were the same as the tools for editing photos. One or the other (or both) would suffer since the tools are dealing with quite different content. Now imagine if you were creating an application for sending money to your friends, and compare it to the application for sending photos to them. You would have to design them to be quite different. Now what if you were designing an interface that was dealing with self service healthcare tools for the elderly; would they be the same as if they were being designed for parents of new-borns?

Usability

- **Usability determines the extent to which a product satisfies:**
 - A user's ability to achieve a stated goal
 - In an effective and efficient manner
- **Every User Experience Designer needs to focus on Usability,**
 - Poor usability will undermine everything else about the design
 - it is not possible to design a good experience if it frustrates the user
- **To understand Usability you will need to think about**
 - Cognitive Limitations
 - Content Considerations
 - User Psychology

10

User Psychology: Users scan screens based on past experiences and expectations. They also identify objects by recognising patterns, so maintain consistency. People believe that things that are close together belong together, so make good use of whitespace. Colour meanings vary by culture, so do not rely on colour alone. Comprehension is not the same as reading; error dialogues are often readable but incomprehensible. Reading speed is quicker with long line lengths, but people prefer short line lengths. Danger, sex, movement, faces and stories get the most attention. People are more motivated the closer they are to the goal. People are addicted to dopamine, which they get from seeking information. Intrinsic rather than extrinsic rewards are the best motivators. People are inherently lazy. People only look for shortcuts if they are easy. People are motivated by autonomy.

User Centred Design (1)

- **User Centred Design is a type of user interface design process**
 - The wants, needs, and limitations of the end users are key
 - Given extensive attention at each stage of the design process
- **It is a multi stage problem solving process and philosophy**
 - Based on the needs and interests of the user
 - With an emphasis on making products usable and understandable



Read “The Design of Everyday Things” by Donald A Norman for more information

11

Design should:

- Make it easy to determine what actions are possible at any moment by employing clear and obvious constraints
- Make things visible, including the conceptual model of the system, the alternative actions, and the results of those actions
- Make it easy to evaluate the current state of the system
- Follow natural mappings between intent and the required actions, the actions and results, and between the information displayed and the interpretation of the system state

User Centred Design (2)

A User Centred Design should:

- **Make it easy to determine what actions are possible at any moment**
 - By employing clear and obvious constraints
- **Make things visible, including:**
 - the conceptual model of the system
 - the alternative actions
 - the results of those actions
- **Make it easy to evaluate the current state of the system**
- **Follow natural mappings between intent and the required actions**
- **The actions and results**
- **Between the information displayed**
- **And the interpretation of the system state**

12

In other words; make sure that

- The user can figure out what to do
- The user can tell what is going on

A UCD Design should exploit natural relationships and constraints

- Operate without instructions or labels in so much as is possible
- Ensure that any instruction or training that is necessary is needed only once

Design should

- Exploit natural relationships and constraints
- Operate without instructions or labels in so much as is possible
- Ensure that any instruction or training that is necessary is needed only once - as simple as possible

Designing in a UCD Manner (1)

To go about the task of creating a UCD design you should:

- Draw upon your and others knowledge and experience
 - Develop a conceptual model appropriate for the user
 - That captures the important parts of the product's operation
 - e.g. its state is readily observable and is easily understood
- Make things visible
 - Show information that allows them to understand and make decisions
 - Provide visual queues to ensure users know the direction of work
 - e.g. a visible control that indicates what a user should do at a glance

13

When going about the task of creating a UCD design:

- Draw upon your knowledge and experience as well as the knowledge and experience of others
 - Develop a conceptual model appropriate for the user that captures the important parts of the product's operation (e.g. its state is easily and readily observable) and is easily understood
- Simplify the structure of tasks
 - Bear in mind that there are limits to how much a person will remember
 - Even if you keep the task the same you can provide mental aids
 - Reminders, alarms, notifications
 - Even if you keep the task the same, automation can reduce the effective complexity for the user
 - Automatic updates, internet time, Automatic daylight saving time adjustments, automatic gear shifts, autopilot, cruise control
 - Display (or communicate) information that would otherwise have been invisible, that provides feedback to the user
 - The “click” sound when the computer registers your click on a button, a fuel gauge, the volume icon on your display, force feedback and rumble in game controllers
 - Change the nature of the task
 - 24 hour digital clocks; is that 12 for midnight or midday? Velcro fasteners

Designing in a UCD Manner (1)

To go about the task of creating a UCD design you should:

- **Draw upon your and others knowledge and experience**
 - Develop a conceptual model appropriate for the user
 - That captures the important parts of the product's operation
 - e.g. its state is readily observable and is easily understood
- **Make things visible**
 - Show information that allows them to understand and make decisions
 - Provide visual queues to ensure users know the direction of work
 - e.g. a visible control that indicates what a user should do at a glance

14

- Make things visible
 - Show people the information that allows them to understand and make the right decisions
 - If a device needs to point in a certain direction to work, then make sure that there is something visible that indicates this to the user at a glance
 - Make sure that something decorative, or a non-core function does not have greater visibility than something with core functionality. E.g. If an aerial needs to be pointed in a certain direction, make sure that the physical design clearly indicates that it is pointing in one direction and does not include anything that could conceivably be another indicator for direction, even if it would look nicer. If a design has what looks like a light or an LED on it, make sure that the light/LED actually functions, and that it works as a self-explanatory status indicator.

Designing in a UCD Manner (2)

- **Simplify the structure of tasks**

- There are limits to how much a person will remember
- Even if you keep the task the same you can provide mental aids
 - Reminders, alarms, notifications
- Even if you keep the task the same, automation can reduce the effective complexity for the user
- Display information that would have been invisible
 - Providing feedback to the user
- Change the nature of the task
 - 24 hour digital clocks; is that 12 for midnight or midday?
 - Velcro fasteners

15

Early and continual focus on users and their tasks

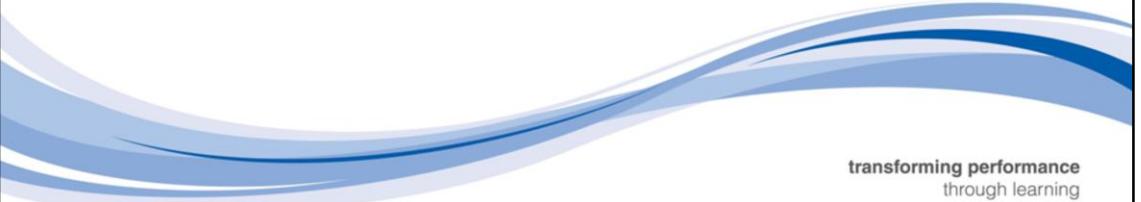
- Do not assume that you know how your customers will use your product. Ask them how they will use it.
- Do not assume that the way your customers use your product is consistent. Keep on checking!

Review

- **The UX Problem**
 - Consumerised users are changing the way we must build
 - Great products offer great interaction and experience
- **Defining UX**
 - Usability, UX & UCD are related methodologies and philosophies
 - They form the basis of our study for this course
- **Starting to think about methodologies**
 - At its core UX is about making the users experience easier
 - We have started thinking about how to do this
 - Now time to apply it to a methodology



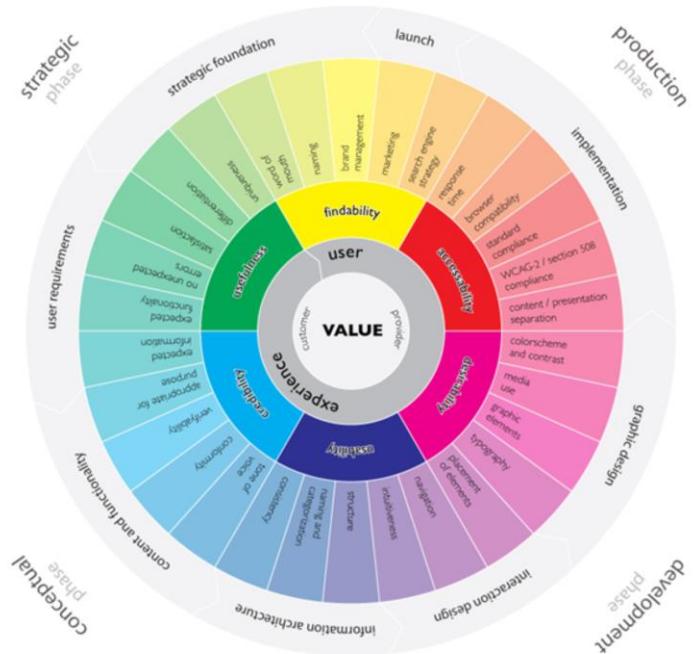
Design Processes & Methodologies



A decorative graphic at the bottom of the slide features several thin, flowing blue lines of varying shades that curve from left to right across the page.

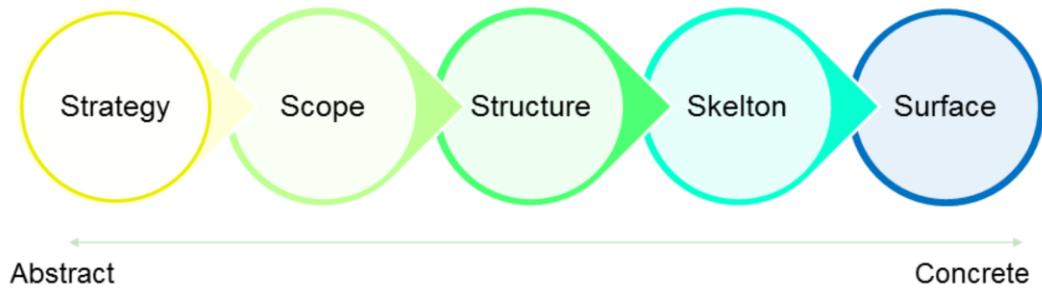
transforming performance
through learning

The UX Wheel



Taming User Experience - The Five Plane Model

- **The five plane model is a way of capturing user information**
 - Ensuring no aspect of the users experience occurs without intent



19

Developed by Garrett Jessie James, the five plane model is a UCD approach very well suited to understanding good UX principles. It builds five planes of activity into the development process for UX starting from the most abstract user centred to a working prototype at the end of development.

The Surface Plane

- **The surface plane is the tactile and visible interface**
 - It involves the physical assets such as colour, fonts & images
- **It is the plane which engages in sensory design**
 - Presenting logical arrangement of information
 - Enhancing the skeleton through concrete appearance decisions
- **The surface is about creating high fidelity wireframes**
 - Widgets
 - Style guides
 - Modules for design
- **It provides the basis of a style guide for physical design**

The Skeleton Plane

- **The skeleton plane defines the form and functionality**
 - Addressing more concrete issues of presentation at an abstract level
 - The high fidelity nature of colours etc... occur in the surface plane
- **The skeleton is defined through:**
 - Interface design – buttons, fields and the like
 - Navigation design – effective journeys through the application
 - Information design – presentation of information in layout
- **These areas are linked together but best considered separately**
 - Then linked together

The Structure Plane

- **The structure plane defines the way in which features fit together**
 - It is a shift from the more abstract information gathering phases
 - Moving towards concrete functionality
- **Also known as interaction design and information architecture**
 - We consider the organisation, grouping and categorising of content
 - Aiming to define patterns and sequences to presented to users
 - It deals with the way we convey information to users

The Scope Plane

- **The Scope plane takes finding of the Strategy plane and aligns them**
 - It ensures we know what we are building
 - Aligning our expectations against the user's
- **Requirements allow us to lay out the deliverables and objectives**
 - Setting out the entire scope mapped out for the project
- **Establish what you are building – ensure everyone understands**
 - Identify requirements, ensure they match user needs
 - Not possible user needs
 - Set out a timeline and stop scope creep
- **We move from why are we making this product to:**
 - What are we going to make

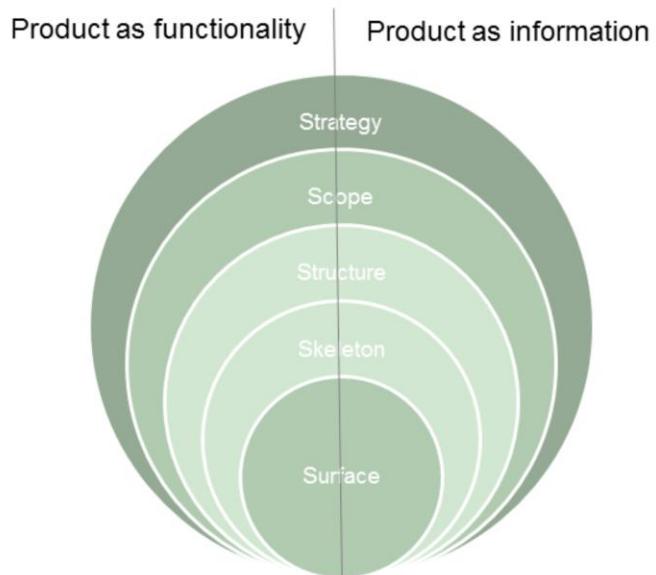
23

Based on evidence we now need to clearly set out our objectives for the development of the product – our evidence from the Strategy plane should provide significant evidence to support or deny early assumptions. The final product must stop being a nebulous construct in a senior stakeholders mind and reach a concrete framework to take towards development.

The Strategy Plane

- **The strategy plane determines initial scope**
 - It defines what the developer and users want from the product
- **It is summed up by two simple questions:**
 - What do we want to get out of this product?
 - What do our users want to get out of it?
- **The answer give us the product objectives & user needs**

Build from the bottom up



Case Study – Social Cinema Booking



A large cinema chain which to use mobile devices to fill the gaps in the cinema goers experience. They are conducting UX research to understand the problem so far they know:

- Users experience in booking, visiting and interacting with the cinema is disconnected.
- A significant set of users utilise social networking, the cinema wants the 'lobby time' included in the design
- Users search elsewhere from the cinema website to find trailers, reviews and opinions
- The cinema wants to 'add value' to the cinema goers experience

Case Study – The Mid Range Hotel

An international hotel chain has taken the decision to update and modernise their website.



- The basic design has not changed in the last five years
- It has approximately 800 hotels across the world
- Despite having an online booking system, most bookings take place through the hotels themselves either by customers phoning, or turning up at the desk



Common Methodologies



A decorative graphic at the bottom of the slide features five horizontal, wavy lines in shades of blue and white, creating a sense of motion and depth. To the right of this graphic, the text 'transforming performance' is written in a bold, sans-serif font, with 'through learning' in a smaller, regular weight font below it.

transforming performance
through learning

Agile Software Development



- If you have not already worked in an “Agile” environment, you will almost certainly have heard of it
- Agile Software Development is a group of software development methods based on iterative and incremental development
- It focuses on adaptive planning, delivery, and development
- It tends towards refinement through evolution



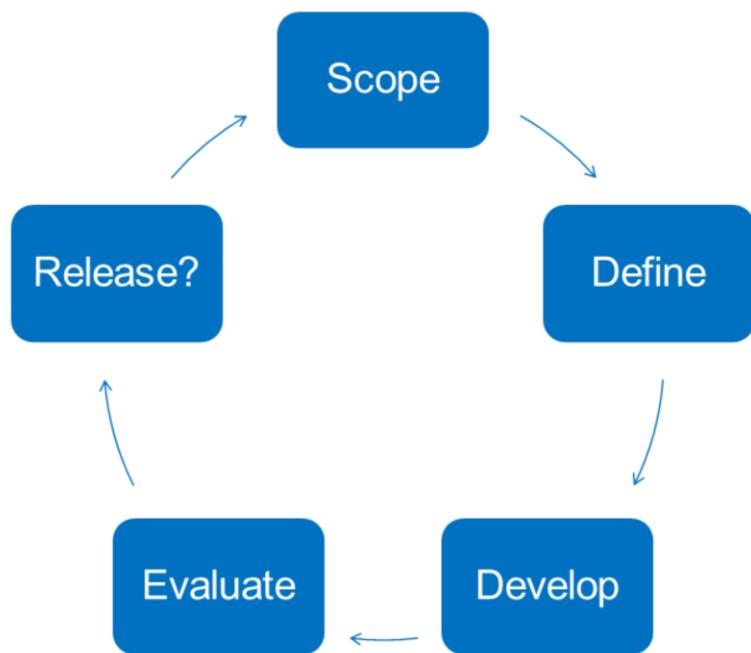
Common Agile Methodologies

Agile Unified Process (Agile UP)
Kanban
Lean Software Development
Rapid Iterative Testing and Evaluation (RITE Method)
Scrum

29

These are some of the common (but by no means all) Agile Methodologies that you are likely to encounter.

Agile Software Development



Waterfall Method



- **The Waterfall model is a sequential design process where progress is seen to be flowing steadily downwards**
- **It has its roots in the manufacturing and construction industries**
- **Is often used as an example of flawed model for software development, but is not without its supporters**
- **It focuses on big design decisions being made up front**

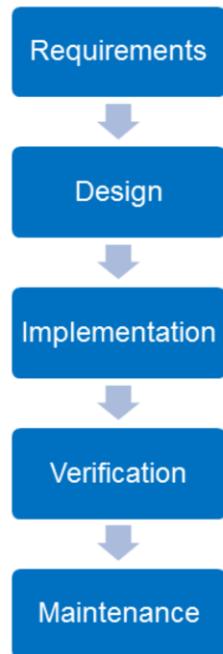
31

Waterfall is useful to understand, even if it leads to problems more often than Agile.

Supporters of Waterfall argue that time spent designing is a worthwhile investment, and conclude that less time is spent fixing bugs found in the early stages of a software product's lifecycle than if it is found late in the cycle. i.e. if there is a problem with the design at the requirements phase, it is easier to fix it then, than the redesign around the problem at the implementation phase and reduces the amount of work that must be scrapped.

Like Agile, is susceptible to “Specification Creep” but a project is more easily derailed with specification creep in Waterfall as there is no short iterative release cycle.

Waterfall Method





Creating a Design-Friendly Process



A decorative graphic at the bottom of the slide features four parallel, flowing blue lines that curve from left to right, creating a sense of motion and fluidity.

transforming performance
through learning

Showing Value

- **The most effective way of getting corporate agreement on a change to the design process is to show the value of it**
- **This is not to say that they will agree to it**
- **Anticipate stumbling blocks**
- **There will be times when you are unable to convince stakeholders**

34

The most effective way of getting corporate agreement on a change to the design process is to show the value of it.

- How much more efficient is it?
- How much money does it save?
- The company needs to get this done quickly; how fast can you do the task with the method you are proposing?

This is not to say that they will agree to it

- There may be other factors at play such as an unwillingness to spend the time on documentation or training changes

Anticipate stumbling blocks

- Provide solutions to negate them where possible

There will be times when you are unable to convince stakeholders

- Wash your hands of the decision, and free your ego from the situation
- Where possible use the results of the project as evidence to support your future proposals

Avoiding Stagnation



- Consider your design process to be a tool that has to be designed.
- Periodically revisit the design of your process
- Analyse what areas if any are not performing as well as they could
- Incorporate changes in education or expertise into your process



Usability Maxims

transforming performance
through learning

Usability

- **The building blocks of Usability are called the 5Es;**
 1. Effectiveness
 2. Efficiency
 3. Engagement
 4. Error Tolerance
 5. Ease of Learning

37

The building blocks of Usability are called the 5Es;

1. Effectiveness
How completely and accurately the task is completed or the goals are reached
2. Efficiency
How quickly the task can be completed
3. Engagement
How effectively the interface draws the user into the interaction and how pleasant and satisfying it is to use
4. Error Tolerance
How well the product prevents errors and helps the user recover from mistakes
5. Ease of Learning
How well the product supports both initial and continued learning

Usability

- When designing a product you will need to study the weighting of each of the 5Es and how they are reflected in the requirements for each of the user groups that you will be designing for
- Their relative importance is determined by their context
- Depending on the situation, some of the dimensions are always more important than others

38

Depending on the situation, some of the dimensions are always more important than others.

E.g. a sales till will always require effectiveness to be a much higher priority than Engagement, but depending on other factors such as the expected turnover of staff, or the complexity of the options available on the products being offered, may put a higher focus on efficiency or error tolerance over the Ease of Learning.

Usability

- It is tempting to say that everything is of equal priority, and an argument can sometimes be made for multiple dimensions to be considered equal priority, but it should only be accepted infrequently



Take ten minutes and decide on a user type, and Prioritise the 5Es for the following scenarios

- Fast Food Till
- Customer Relationship Management Suite
- Social Media Application
- Government Information Website
- Interactive news media Website
- Free to play web game

Write a quick justification for each of your decisions



Lund's Usability Maxims



A decorative graphic at the bottom of the slide features five parallel, flowing blue lines of varying shades, creating a sense of motion and depth.

transforming performance
through learning

Lund's Usability Maxims

- Arnold Lund published his Usability Maxims in 1997 in the Ergonomics in Design journal
- The following 34 Maxims are in order of importance, from least important to most important according to Lund
- These are not a usability design gospel, and must be adapted to your own needs and process

Lund's Usability Maxims

34. **To know the system is to love it**
33. **Let people shape the system to themselves, and paint it with their own personality**
32. **Cute is not a good adjective for systems**

42

34. To know the system is to love it.

When a user knows the ins and outs of the software, they should feel good about using it as it enables them to achieve their goals.

33. Let people shape the system to themselves, and paint it with their own personality.

The software should be flexible enough to work with the user's own workflow.

32. Cute is not a good adjective for systems.

You want it to work well and efficiently, not childishly or in a quaint manner.

Lund's Usability Maxims

31. If I made an error, at least let me finish my thought before I have to fix it
30. The user should be in a good mood when done
29. Everything in its place, and a place for everything

43

31. If I made an error, at least let me finish my thought before I have to fix it.
Try not to be too intrusive, especially when it comes to warnings and errors.
30. The user should be in a good mood when done.
A user who is aggravated by the software will not want to use the software and avoid it as much as possible.
29. Everything in its place, and a place for everything.
Make sure that everything can be found easily and logically.

Lund's Usability Maxims

28. Colour is information
27. If it is not needed, it's not needed
26. The fault is not in yourself, but in your system

44

28. Colour is information.
Do not use colours that can be seen as meaning something specific, except in those instances when it does.
27. If it is not needed, it's not needed.
Simplicity, simplicity, simplicity. Do not clutter the experience with unnecessary features.
26. The fault is not in yourself, but in your system.
Do not hold on to a design element as beloved. If it doesn't make the cut, it is not a reflection of you or your abilities. It's just not right for that particular context.

Lund's Usability Maxims

25. Provide a way to bail out and start over
24. Keep it neat. Keep it organised
23. Design for regular people and the real world

45

25. Provide a way to bail out and start over

Make it easy for the user to decide that it's easier to start again from scratch.

Whether it's because they've finished what they were looking to do, or because they've made a mistake is immaterial.

24. Keep it neat. Keep it organised.

If it's not neat and organised, the user will have difficulty finding things.

If the user cannot find something, they will assume that it isn't there and feel aggravated. If it is pointed out to them, they may feel aggravated **and** foolish.

23. Design for regular people and the real world.

Do not design for the "ideal" user. Design for the people who will actually use your software.

Lund's Usability Maxims

22. Even experts are novices at some point. Provide help
21. Don't let people accidentally shoot themselves
20. You should always know how to find out what to do next

46

22. Even experts are novices at some point. Provide help.
Nobody always remembers every step of a complex system. Guide them with help files and wizards.
21. Don't let people accidentally shoot themselves.
If it doesn't work, why can they do it?
Don't use checkboxes when a radio selector would be more appropriate etc.
20. You should always know how to find out what to do next.
Make sure that your design makes it easy to get to the help files, or that it guides the user to the next steps.
Your usability concerns should extend to the documentation.

Lund's Usability Maxims

19. **Things that look different should act different**
18. **The user should be able to do what the user wants to do**
17. **The best journey is the one with the fewest steps. Shorten the distance between the user and their goal**

47

19. Things that look different should act different.
If it looks like something else, it should act like it, or users will be frustrated by it acting in an unexpected manner.
18. The user should be able to do what the user wants to do.
People use our software to make achieving their goals easier. If we can't do this, we are failing our customers.
17. The best journey is the one with the fewest steps. Shorten the distance between the user and their goal.
Try to never be more than two clicks away from where the user needs to go next.

Lund's Usability Maxims

16. If I made an error, let me know about it before I get into REAL trouble
15. Eliminate unnecessary decisions, and illuminate the rest
14. The idea is to empower the user, not speed up the system

48

16. If I made an error, let me know about it before I get into REAL trouble.
This is why they tell you the bridge is out before you get there. Not once you get to the other side.
15. Eliminate unnecessary decisions, and illuminate the rest.
If it adds no real value, then remove it. If it adds real value, then promote it.
Focus on the decisions which are important to the user.
14. The idea is to empower the user, not speed up the system.
The user is the focus. Designing for a faster workflow is only good if that is what the user is looking for. Do not sacrifice their needs in the name of streamlining

Lund's Usability Maxims

13. The user should control the system. The system shouldn't control the user. The user is the boss, and the system should show it
12. The user should always know what is happening
11. The more you do something, the easier it should be to do

49

13. The user should control the system. The system shouldn't control the user. The user is the boss, and the system should show it.

If your user doesn't feel like they are in control, they will not feel that the software is helping them achieve **their** objectives.

12. The user should always know what is happening.

A user who does not know what is happening is a user who does not feel in control. A user not in control does not feel that the software is a worthwhile use of their time and effort and uses it only begrudgingly.

11. The more you do something, the easier it should be to do.

The more often the user does something, the more opportunity there is for aggravating them.

Lund's Usability Maxims

10. **Keep it simple**
9. **Minimise the need for a mighty memory**
8. **Consistency, consistency, consistency**

50

10. Keep it simple.

“Perfection is reached not when there is nothing left to add, but when there is nothing left to take away” (Antoine de Saint Exupéry)

“Simplicity is the ultimate sophistication” (Leonardo Da Vinci)

“Everything should be made as simple as possible, but no simpler” (Albert Einstein)

9. Minimise the need for a mighty memory.

If the user needs to remember a lot of information just to be able to use the tool, they will forget things and make regular mistakes

8. Consistency, consistency, consistency.

There is no such thing as a truly intuitive software interface. Only an easily learned one. It is difficult to learn something if it doesn't follow logical rules.

Lund's Usability Maxims

7. **Don't overwhelm the user**
6. **Every action should have a reaction**
5. **Error messages should actually mean something to the user, and tell the user how to fix the problem**

51

7. Don't overwhelm the user.

Too much data, or too many options create analysis paralysis. It's easier to chose from three options than from thirty.

6. Every action should have a reaction.

The user needs to know that whatever action it is that they've taken as been recognised. Visual feedback is a must.

5. Error messages should actually mean something to the user, and tell the user how to fix the problem.

If you have nothing helpful to say, don't say it.

When we don't have the option of not saying something, we'd better tell them something helpful.

The first Wing Commander game had an error message that occurred every time the player quit the game. They couldn't find the cause of the error in time, and it didn't negatively effect the game, so instead they edited the error message to say "Thank you for playing Wing Commander"

Lund's Usability Maxims

4. The information for the decision needs to be there when the decision is needed
3. Everyone makes mistakes, so every mistake should be fixable
2. Things that look the same should act the same

52

4. The information for the decision needs to be there when the decision is needed.

It's no good asking for a decision on something when the data needed for that decision is only available somewhere else.

3. Everyone makes mistakes, so every mistake should be fixable.

The ability to undo a mistake, by whatever means, without losing work is a must.

2. Things that look the same should act the same.

You've taught the users that certain visual cues mean certain things. Don't infuriate them by making that inconsistent.

Lund's Usability Maxims

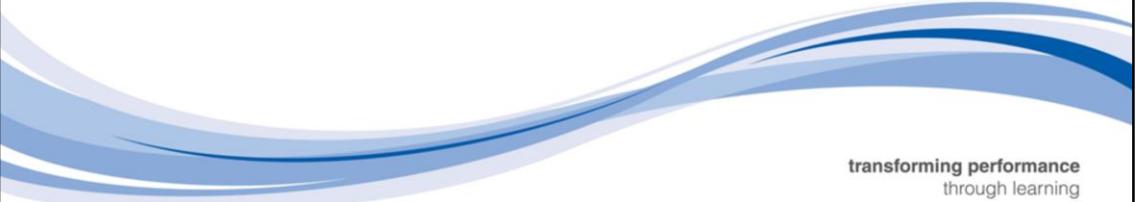
1. Know the user, and you are not the user

53

Theory is all well and good, but you need to put it in front of real users.



Jakob Nielsen's Ten Usability Heuristics



A decorative graphic at the bottom of the slide features five horizontal, wavy lines in shades of blue and white, resembling water or a stylized ribbon.

transforming performance
through learning

Nielsen's Usability Heuristics

- **Visibility of system status**
- **Match between system and the real world**
- **User control and freedom**

55

Visibility of system status

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

Match between system and the real world

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.

User control and freedom

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.

Nielsen's Usability Heuristics

- **Consistency and standards**
- **Error prevention**
- **Recognition rather than recall**

56

Consistency and standards

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

Error prevention

Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

Recognition rather than recall

Minimise the user's memory load by making 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.

Nielsen's Usability Heuristics

- **Flexibility and efficiency of use**
- **Aesthetic and minimalist design**
- **Help users recognise, diagnose, and recover from errors**
- **Help and documentation**

57

Flexibility and efficiency of use

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.

Aesthetic and minimalist design

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.

Help users recognise, diagnose, and recover from errors

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

Help and documentation

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

Review

- **Usability, UX and Information Architecture are related**
 - They provide a user focused way to consider product design
- **A methodological approach to UX is essential**
 - Without structure you are destined to fail
- **UX can be applied into software engineering processes**
 - Getting buy in matters