Ten Usability Heuristics (Nielsen, 2001)

1. Visibility of System Status	The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.
2. 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.
3. 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.
4. Consistency and Standards	Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.
5. Error Prevention	Even better than good error messages is a careful design which prevents a problem from occurring in the first place.
6. Recognition rather than Recall	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.
7. 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.
8. 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.
9. 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.
10. 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.

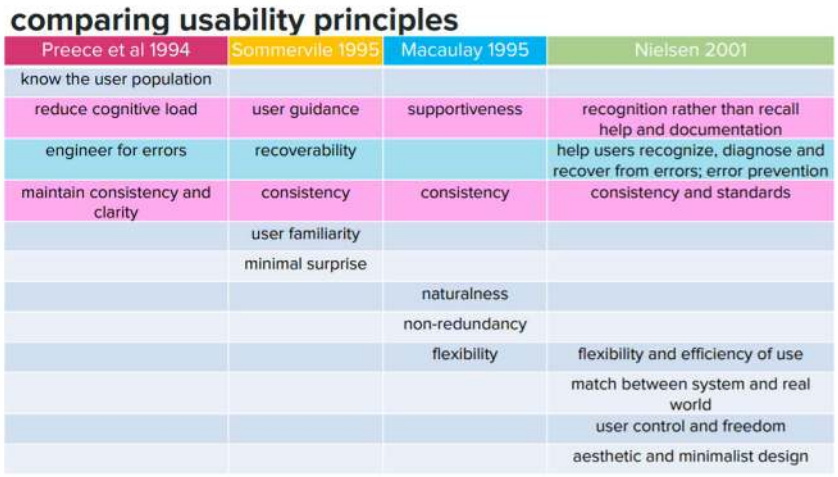
Immersion is the process whereby you become 'immersed' or deeply interested in a story or particular material. Immersion implies a passive act, rather than something you are actively **engaged** in. Engagement is the process where you are actively 'engaged' in solving a particular problem. Engagement implies action – like trying to overcome a challenge, understand some difficult material, or solve some conundrum.

Immersion vs engagement. For example in games, immersion is typically about the story whereas engagement is about the game play. Certainly, games can have both immersion and engagement. But, a game without engagement is hardly a game at all. And, sometimes, the distinction is not always so clear. For instance, you could become deeply engaged trying to unravel the complexities of a story.

Usability Design Principles (Preece et al, 1994)	
Principle	**Description**
Know the User Population	Being sympathetic to different user needs
Reduce the Cognitive Load	Design so that users don't have to remember large amounts of detail
Engineer for Errors	People always make mistakes. The system can be designed both to prevent errors and to enable recovery from errors
Maintain Consistency and Clarity	From standard operations and representations and from using appropriate metaphors. A designer's understanding of what is 'clear' will depend on their understanding of their users

Usability Design Principles for Software Engineering (Sommerville, 1995)	
Principle	**Description**
User familiarity	The interface should use terms and concepts which are drawn from the experience of the anticipated users
Consistency	The interface should be consistent in that, comparable operations should be activated in the same way
Minimal Surprise	Users should never be surprised by the behaviour of a system
Recoverability	The interface should include mechanisms to allow users to recover from their errors
User Guidance	The interface should incorporate some from of context sensitive user guidance and assistance

Usability Design Principles (Macaulay, 1995) Human-Computer Interaction for Software Designers	
Principle	**Description**
Naturalness	The user does not have to alter their approach to their work in order to interact with the system
Consistency	Expectation built up through the use of one part of the system are no frustrated by changed in another
Non-redundancy	The user needs to input only the minimum information for the systems operation
Supportiveness	The 'dialogue' assists the user to use the system
Flexibility	Different levels of user familiarity should be supported



1. **Visibility**
   - The more visible functions are, the more likely users will be able to know what to do next. In contrast, when functions are “out of sight”, it makes them more difficult to find and know how to use.

2. **Feedback**
   - Provides acknowledgement of our interactions, and information about their outcomes.
   - Comes in many forms, selected state, highlights, dialogs, tool tips, confirmation and error messages, sounds, page refreshes, loading. Subtle: breadcrumbs that tell us where we are on sites.
   - It should confirm the interaction, and the outcome, and if its important, actions should be verified.

3. **Constraints**
   - The design concept of constraining refers to determining ways of restricting the kind of user interaction that can take place at a given moment. There are various ways this can be achieved.

4. **Mapping**
   - This refers to the relationship between controls and their effects in the world. Nearly all artifacts need some kind of mapping between controls and effects, whether it is a flashlight, car, power plant, or cockpit. An example of a good mapping between control and effect is the up and down arrows used to represent the up and down movement of the cursor, respectively, on a computer keyboard.

5. **Consistency**
   - Users will learn faster how to use your design. It eliminates confusion! *easier to learn and use*
   - Is it consistency in terms of:

     • appearance (looks the same)
     • the visual metaphors, icons, elements used etc
     • in steps to accomplish similar interactions
     • standard conventions

   - Components with similar behaviour should have a similar appearance
   - Components with different behaviour should have a different appearance

6. **Affordance**
   - is a term used to refer to an attribute of an object that allows people to know how to use it. For example, a mouse button invites pushing (in so doing acting clicking) by the way it is physically constrained in its plastic shell. At a very simple level, to afford means "to give a clue" (Norman, 1988). When the affordances of a physical object are perceptually obvious it is easy to know how to interact with it.

7. **Signifier**
   - If, affordances allow people to perceive what actions are possible, then, Signifiers conveys where the action should take place
   - Signifiers are signals, e.g. signs, labels, and drawings. E.g., “push,” “pull,” or “exit” on doors, or arrows and diagrams indicating what is to be acted upon or in which direction to gesture, or other instructions.
   - Provide some sign of what it is for, what is happening, and what the alternative actions are.