5.1. USABILITY TESTING

1. What is Usability Testing?

Usability Testing is the process of testing a website or application with real users to check whether it is easy to use.

Example: Asking users to search for a product, add it to the cart, and place an order.

2. Why is Usability Testing Important?

- Finds design problems.
- Makes applications easy to use.
- Improves user experience.
- Saves time and money.
- Reduces user confusion.
- Improves customer satisfaction.

3. Goals of Usability Testing

1. **Effectiveness:** Can users complete tasks?
2. **Efficiency:** How quickly can users complete tasks?
3. **Satisfaction:** Are users happy with the application?
4. **Learnability:** Is it easy for beginners?
5. **Accessibility:** Can everyone use it?
6. **Error Prevention:** Does it prevent mistakes?

4. Key Usability Attributes

Attribute	Meaning
Effectiveness	Complete tasks correctly
Efficiency	Complete tasks quickly
Learnability	Easy to learn
Memorability	Easy to remember
Error Prevention	Avoid mistakes
Satisfaction	User feels comfortable
Accessibility	Usable by everyone

5. Usability Testing Process

1. **Define objectives:** Testing objectives explain what we want to test.

   Example: Check whether users can easily search and order food using a food delivery application.

2. **Select target users:** Target users are the people who will use the application.

   Example:

   - Students
   - Teachers
   - Working professionals
   - Beginners

3. **Prepare tasks:** Test tasks are activities given to users.

Example:

- Search for a product.
- View product details.
- Add to cart.
- Apply a coupon.
- Place an order.

4. **Create test scenarios:** A test scenario is a real-life situation given to users for testing.

   Example: You want to buy a mobile phone. Open the shopping app, search for a phone, and add it to your cart.

5. **Select testing method:**
   - **Moderated Testing:** A person guides the user.
   - **Unmoderated Testing:** User tests independently.
   - **Remote Testing:** Testing happens online.
   - **In-Person Testing:** Testing happens face-to-face.
6. Conduct testing.
7. Observe users.
8. Collect feedback.
9. Find problems.
10. Improve the design.
11. Test again.

Severity of Usability Problems:

Level	Meaning
1 – Minor	Small problem
2 – Moderate	Noticeable problem
3 – Major	Serious problem
4 – Critical	Task cannot be completed

Usability Testing – Detailed Notes

Understand users • Find problems • Improve experiences

1. Introduction to Usability Testing

Usability testing is a UX **research method** used to evaluate how easily and effectively users can use a product, website, mobile application, or interactive system.

In usability testing, **real users perform realistic tasks** while researchers observe their behavior, identify problems, and collect feedback.

Simple Definition:
Usability testing is the procees of observing users **while** they perform tasks on a product to identify usability problems and improve the user experience.

Example: Online Pharmacy App

Test tasks:

1. Search for a medicine.
2. Check its details.
3. Add it to the cart.
4. Apply a coupon.
5. Place an order.



If several users cannot find the **Add to Cart** button, the design has a usability problem.

2. Why Usability Testing is Important

Helps designers **understand** whether the product works well from the user's point of view.

Major benefits

- Identifies usability problems.
- Reduces user confusion.
- Improves navigation.
- Improves task completion.
- Saves development costs.
- Improves accessibility.
- Increases user satisfaction.
- Validates design decisions.
- Discovers unexpected user behavior.
- Provides evidence for design improvements.

Design process comparison

Without testing:

Designer → Assumption → Design → Development

With usability testing:

Designer → Prototype → User Testing → Problems → Improvements → Final Design

3. Goals of Usability Testing

Determine whether users can successfully use the system.

Goal	Example
Effectiveness	Can the user book a bus ticket?
Efficiency	How long does it take to book a ticket?
Satisfaction	Was the process easy to understand?
Learnability	How easily new users understand the interface?
Error Prevention	Does the interface help prevent and recover from errors?
Accessibility	Can people with different abilities use the product?

4. Key Usability Attributes

A usable system has the following characteristics:

Attribute	Meaning	Example
Effectiveness	Complete tasks correctly	Submit a form
Efficiency	Fast completion	Fast checkout
Learnability	Easy for new users	Clear navigation
Memorability	Remember after a gap	Familiar icons
Error Prevention	Prevents mistakes	Confirm before delete
Error Recovery	Helps recover	Undo option
Satisfaction	Feels comfortable	Simple interface
Accessibility	Usable by all	Screen-reader support

5. Usability Testing Process

1. Define objectives
2. Identify target users
3. Prepare test tasks
4. Create test scenario
5. Select testing method
6. Recruit participants
7. Conduct test
8. Observe & record
9. Analyze results
10. Identify problems
11. Prioritize issues
12. Improve design
13. Retest



6. Step 1 – Define Testing Objectives

Decide what needs to be tested.

Example: Food delivery app

Objective: Evaluate if users can search for a restaurant, select food items, and complete checkout easily.

Good objectives: Specific, measurable, related to user goals, important features.

Poor objective

Test the app.

Better objective

Determine whether first-time users can place a food order without assistance.

7. Step 2 – Identify Target Users

Participants should represent actual or intended users.

Example: Online learning platform

- College students
- Teachers
- Working professionals
- First-time users



Important characteristics

- Age group
- Technical experience
- Domain knowledge
- Frequency of use
- Accessibility requirements
- Device preferences

Example: College attendance app

Target users: Students, Faculty, Administrators

Testing only developers is not representative.

8. Step 3 – Prepare Test Tasks

Use realistic, goal-oriented tasks.

Example: E-Commerce application

1. Find a smartphone under ₹20,000.
2. Compare two products.
3. Add one product to the cart.
4. Apply a discount coupon.
5. Proceed to checkout.

Good task characteristics

- Realistic
- Clear
- Goal-oriented
- Independent
- Easy to understand

Avoid leading instructions

✗ "Click the blue shopping-cart icon..."

✓ "Add the phone you selected to your shopping cart."

9. Step 4 – Create a Test Scenario

Provides context for the task.

Example: "You are planning a trip to Chennai next weekend. Use the travel application to find a suitable bus and proceed up to the payment page."



Makes the task feel realistic.

10. Step 5 – Select Testing Method

Moderated
Researcher present. Pros: detailed, Q&A. Cons: time, cost, may influence.

Unmoderated
Independent. Pros: many users, low cost. Cons: no immediate help, less qualitative.

Remote
Different locations. Pros: user devices, geographic reach. Cons: less observation.

In-Person
Same location. Pros: rich observation, body language. Cons: higher cost.

11. Formative vs Summative

Formative Testing

- During design process
- Find problems early
- Example: test a low-fidelity prototype

Summative Testing

- After design is mature
- Measure usability
- Metrics: success rate, time, errors, satisfaction

Comparative Testing

Test two or more designs and compare results.
Example: bottom navigation vs hamburger menu.

12. Other Testing Methods

Guerrilla Testing

- Quick, low cost, convenient
- Example: test with campus students
- Limitation: may not represent target users.

Remote vs In-Person

Feature	Remote	In-Person
Location	Different	Same
Cost	Lower	Higher
Convenience	High	Moderate
Observation	Digital only	Digital + physical
Geographic reach	High	Limited
Environment control	User's env.	Controlled env.

13. Recruiting Participants & Test Environment

Participants

Example (LMS):

Example 1	Beginner
Student 1	Beginner
Student 2	Intermediate
Student 3	Frequent user
Student 4	Occasional user
Student 5	Beginner

Number depends on objective, complexity, study design, and confidence in findings.

Test Environment

Required elements:

- Device
- Application/prototype
- Internet (if needed)
- Test tasks
- Consent information
- Note-taking
- Screen recording (if permitted)
- Audio recording (if permitted)
- Observation sheet

14. Test Moderator

Guides the session.

Responsibilities:

- Make participants comfortable
- Provide tasks
- Avoid influencing
- Observe behavior
- Ask neutral questions
- Record observations
- Keep the session focused

Test the design, not the user.
Investigate what in the interface caused the mistake.

15. Think-Aloud & Observation

Think-Aloud Method

Users describe what they are thinking.

Example: Grammar

Example: "I want to find my orders, but I don't see where they are."

Helps understand: expectations, confusion, mental models, decision-making.

Limitation: may affect natural behavior.

Observation

Observe what users actually do.

Watch for: clicks, hesitations, ignored items, errors, scrolling, reactions, questions.

Example: Repeatedly clicking a product image = discoverability problem.

16. Quantitative Metrics

Task Completion Rate

$$\frac{\text{Successful Tasks}}{\text{Total Tasks}} \times 100$$

Ratio ~ 80%

Task Completion Time

Average or median time.

U1	2 min	U2	3 min
U3	2.5 min	U4	4 min

Error Rate

$$\frac{\text{Errors}}{\text{Opportunities}} \times 100$$

12/100 = 12%

Satisfaction

Questionnaires, rating scales, interviews.

1 2 3 4 5

Very Difficult — Very Easy

17. Qualitative Data

Explains why a problem occurs.

Examples: user comments, observations, interviews, behavior, confusion, frustration, expectations.

Example: 40% failed to find settings.
Reason: expected it under profile picture.

Together with quantitative data, it provides stronger insight.

18. Usability Testing Metrics

Performance

- Task completion rate
- Completion time
- Error rate
- Number of clicks
- Attempts
- Abandonment rate

User Experience

- Satisfaction
- Confidence
- Ease of use
- Perceived difficulty
- User comments

Accessibility

- Keyboard navigation
- Screen-reader support
- Text readability
- Color contrast
- Time/mobility

19. Severity of Usability Problems

1 – Minor	Small inconvenience. E.g., unclear label.
2 – Moderate	Noticeable difficulty. E.g., several attempts to find.
3 – Major	Prevents or disrupts task. E.g., hard to find checkout.
4 – Critical	Cannot complete important task. E.g., cannot submit emergency alert.

20. Usability Testing Report

1. Introduction
2. Objectives
3. Participants
4. Methodology
5. Findings
6. Metrics
7. Recommendations
8. Conclusion

21. Example Report

Product: Food Delivery App
Objective: Order food without help
Participants: 10 students

Key findings:

Problem	Users	Severity
Search icon unclear	3	Medium
Coupon hidden	5	High
Cart icon clear	Most	Low
Checkout unclear	2	Medium

Recommendation: Improve search, make coupon easier, clearer checkout.

22. Usability Testing vs User Interview

Usability Testing	User Interview
Perform tasks	Answer questions
Observes behavior	Explores opinions
Finds interaction problems	Understands needs & attitudes
Measures performance	Explores experiences
"Can users do it?"	"What do users think/want?"

23. Usability Testing vs A/B Testing

Usability Testing	A/B Testing
Observes tasks	Compares variants
Why problems occur	Outcome differences
Qual + Quant	Primarily quantitative
During design iteration	When variants can be deployed
Small groups	Larger traffic needed

24. Usability Testing in UX Process

Research → User Req. → IA → Wireframe → Prototype → Testing → Problems → Redesign → Retest → Final Product

Can be repeated several times.

25. Women's Safety App Example

Tasks: SOS alert, add contact, share location.
Test: find SOS, clear instructions, recover, fast interaction, accessible, confirmation.

Pay special attention to error prevention, recovery, discoverability, accessibility.

26. Common Usability Problems

- **Navigation:** don't know where to go
- **Visibility:** hard to see important items
- **Consistency:** different behavior
- **Feedback:** no clear indication
- **Error:** avoidable mistakes
- **Terminology:** unfamiliar terms
- **Accessibility:** not enough support

27. Good Practices

Before

- Clear objectives
- Representative users
- Realistic tasks
- Pilot test
- Observation sheet
- Get consent

During

- Comfortable
- Neutral
- Avoid leading Qs
- Observe carefully
- Encourage thinking
- Record evidence

After

- Organize findings
- Find recurring problems
- Separate facts vs interpretation
- Prioritize by severity
- Recommend changes
- Retest

28. Common Mistakes to Avoid

- ✗ Testing only with designers/devs
- ✗ Revealing the answer
- ✗ Leading questions
- ✗ Focusing only on opinions
- ✗ Ignoring accessibility
- ✗ Testing only after development
- ✗ Treating all problems equally
- ✗ Changing design based on one comment

29. Key Takeaways

- ✓ Evaluates how easily users can use a product.
- ✓ Involves real users performing tasks.
- ✓ Finds problems designers may miss.
- ✓ Key dimensions: effectiveness, efficiency, learnability, memorability, error handling, satisfaction, accessibility
- ✓ Can be moderated, unmoderated, remote, in-person, formative, summative, comparative.
- ✓ Use both qualitative and quantitative data.
- ✓ Prioritize by severity and impact.
- ✓ Lead to redesign and retesting.
- ✓ Observe what users actually do, not only what they say.

5.2. HEURISTIC EVALUATUION

1. What is Heuristic Evaluation?

- **Heuristic Evaluation** is a method of checking a website or application to find usability problems using a set of predefined design principles.
- Experts check an application to find design mistakes and see whether it is easy for users to use.

Example: Checking a shopping app to see whether the buttons are clear, navigation is easy, and error messages are understandable.

2. Why is Heuristic Evaluation Important?

- Finds usability problems.
- Improves user experience.
- Makes navigation easier.
- Reduces user confusion.
- Saves time and cost.
- Improves application quality.

3. 10 Principles of Heuristic Evaluation:

These principles were introduced by usability expert Jakob Nielsen.

1. **Visibility of System Status:**

The application should inform users about what is happening.

Example: Show a loading symbol when a page is loading.

2. **Match Between System and Real World:**

Use simple words and familiar language.

Example: Use a shopping cart icon for purchased products.

3. **User Control and Freedom:** Allow users to undo or cancel actions.

Example: Provide a Cancel or Back button.

4. **Consistency and Standards:** Keep buttons, colors, and layouts consistent.

Example: Use the same button style on every page.

5. **Error Prevention:** Design the application to prevent mistakes.

Example: Show a confirmation message before deleting a file.

6. **Recognition Rather Than Recall:**

Show options so users do not need to remember everything.

Example: Display search suggestions.

7. **Flexibility and Efficiency of Use:**

Make the application easy for beginners and experienced users.

Example: Provide keyboard shortcuts.

8. **Aesthetic and Minimalist Design:**

Keep the design simple and avoid unnecessary information.

Example: Use a clean login page.

9. **Help Users Recognize, Diagnose, and Recover from Errors:**

Show clear error messages and solutions.

Example: "Incorrect password. Please try again."

10. **Help and Documentation:** Provide instructions when users need help.

Example: FAQ section and help page.

4. Steps in Heuristic Evaluation

1. Select the website or application.
2. Select usability experts.
3. Choose the evaluation principles.
4. Experts examine the application.
5. Identify usability problems.
6. Record the problems.
7. Assign severity levels.
8. Suggest improvements.
9. Fix the problems.

5. Severity Levels

Severity means how serious a usability problem is.

Level	Meaning
0	Not a usability problem
1	Cosmetic problem
2	Minor problem
3	Major problem
4	Critical problem

6. Example – Shopping Application

An expert checks a shopping application and finds:

Problem	Heuristic Principle
No loading indicator	Visibility of system status
Confusing icons	Match with real world
No cancel button	User control and freedom
Different button styles	Consistency
No delete confirmation	Error prevention
No search suggestions	Recognition rather than recall
No keyboard shortcuts	Flexibility and efficiency
Too much information	Minimalist design
Unclear error message	Error recovery
No help section	Help and documentation

7. Advantages of Heuristic Evaluation

- Easy to perform.
- Finds design problems quickly.
- Less expensive than many user-testing methods.
- Improves usability.
- Helps improve design quality.
- Can be done before development.

8. Disadvantages of Heuristic Evaluation

- Depends on expert knowledge.
- Different experts may find different problems.
- Does not directly show real user behavior.
- Some user problems may be missed.

Heuristic Evaluation

Identify Usability Problems • Improve User Experience

1. Introduction

Heuristic Evaluation is a usability inspection method used to identify usability problems in a user interface by evaluating it against a set of established usability principles called heuristics.

Unlike usability testing, it does not necessarily require real users. Usually, UX designers, usability experts or trained evaluators inspect the interface systematically.

Definition:
Heuristic evaluation is the systematic inspection of a user interface using established usability principles to identify potential usability problems.

Example: Food delivery application

- Is the current page clearly indicated?
- Does the system show feedback after placing an order?
- Can users easily undo an accidental action?
- Are buttons and labels consistent?
- Are error messages understandable?

2. Purpose of Heuristic Evaluation

The main purpose is to discover usability problems early and improve the interface.

Objectives

- Identify usability problems.
- Improve navigation.
- Reduce user errors.
- Improve consistency.
- Improve system feedback.
- Increase learnability.
- Improve user satisfaction.
- Identify accessibility-related concerns.
- Evaluate prototypes before development.
- Reduce the cost of fixing usability problems later.

Why early?
Fixing a usability problem in the early stages is much cheaper than fixing it after full development.

3. Heuristic Evaluation vs Usability Testing

Aspect	Heuristic Evaluation	Usability Testing
Evaluators	Experts inspect the interface	Real users perform tasks
Usability principles	Uses heuristics	Uses realistic user tasks
Users required	Not necessarily	Requires participants
Problems found	Potential usability problems	Actual user problems
When it can be done	Early (wireframes, prototypes, design)	Usually with a prototype or product
Time	Relatively quick	Can require more time
Focus	Expert judgment	User behavior

Heuristic Evaluation: "Does the interface follow good usability principles?"
Usability Testing: "Can users actually use the interface successfully?"
Both methods complement each other.

4. Nielsen's 10 Usability Heuristics

1. **Visibility of System Status**
   Keep users informed about what is happening through appropriate feedback.
   Example: Loading indicator, progress bar, "Payment successful".
2. **Match Between System and Real World**
   Use language, concepts and conventions familiar to users.
   Example: "Transfer Money" instead of "Initiate financial transaction".
3. **User Control and Freedom**
   Allow users to undo, cancel or go back.
   Example: Undo, Redo, Cancel, Back, Close.
4. **Consistency and Standards**
   Maintain consistency in terminology, colors, icons, buttons and navigation.
   Example: Same button style for similar actions.
5. **Error Prevention**
   Prevent errors before they happen.
   Example: Input validation, confirmation before delete.
6. **Recognition Rather Than Recall**
   Make information and options visible, not-based on memory.
   Example: Dropdown lists, suggestions, recent items.
7. **Flexibility and Efficiency of Use**
   Support both beginners and experienced users.
   Example: Keyboard shortcuts, quick actions, customization.
8. **Aesthetic and Minimalist Design**
   Avoid unnecessary information and focus on what is important.
   Example: Clean layout, proper whitespace.
9. **Help Users Recognize, Diagnose, and Recover from Errors**
   Provide clear error messages and solutions.
   Example: "Password must be at least 8 characters."
10. **Help and Documentation**
    Provide easily accessible help when needed.
    Example: FAQs, tutorials, help center.

5. Heuristic Evaluation Process

1. Select Interface
2. Define Evaluation Goals
3. Select Heuristics
4. Select Evaluators
5. Independent Inspection
6. Record Problems
7. Assign Severity
8. Combine Findings
9. Prioritize Problems
10. Recommend Improvements
11. Redesign
12. Re-evaluate

The process is systematic and helps in identifying and prioritizing usability issues efficiently.

6. Step 1 – Define the Scope

Decide what will be evaluated.

Example: Online pharmacy
- Product details
- Cart • Checkout • Payment • Order tracking

7. Step 2 – Select Evaluators

Evaluators should understand:
- UX principles
- Interface design
- Usability
- Interaction design
- The target domain

Multiple evaluators help find more problems.

8. Step 3 – Independent Inspection

Each evaluator inspects the interface independently and looks for heuristic violations.

Example: No confirmation after payment → Visibility of System Status.

9. Step 4 – Record Findings

Document each problem clearly.

Screen	Problem	Heuristic	Evidence	Evidence
Checkout	No payment progress indicator	Visibility	3	User can't tell if processing.

10. Step 5 – Assign Severity

Use a severity scale to prioritize issues.

Level	Description
0	Not a usability problem
1	Cosmetic (minor)
2	Minor (small problem)
3	Major (significant problem)
4	Critical (prevents important tasks)

Severity is a prioritization aid, not a measure of designer or product quality.

11. Example – Donlla Heuristics App

No.	Heuristic	Main Idea
1	Visibility of System Status	Keep users informed
2	Match with Real World	Use familiar language
3	User Control and Freedom	Provide undo/cancel
4	Consistency and Standards	Maintain consistency
5	Error Prevention	Prevent mistakes
6	Recognition Rather Than Recall	Reduce memory load
7	Flexibility and Efficiency	Support beginners & experts
8	Aesthetic & Minimalist Design	Avoid unnecessary elements
9	Error Recovery	Explain and solve errors
10	Help & Documentation	Provide assistance

Memory Trick: V – M – U – C – E – R – F – A – E – H

12. Example – Food Delivery App

Problems & Recommendations

Screen	Problem	Heuristic	Severity	Recommendation
Search	No loading indicator	Visibility	3	Add loading indicator
Cart	No undo after removal	User Control	3	Show "Undo"
Checkout	Generic error message	Error Recovery	2	Provide specific explanation

13. Example – Online Pharmacy App

Common issues found:
- No loading message during payment (Visibility)
- Technical term "Rx authentication" (Real World)
- No cancel option (User Control)
- Different button styles (Consistency)
- Invalid phone number allowed (Error Prevention)
- Must remember medicine code (Recognition)
- No search suggestions (Flexibility)
- Crowded product page (Minimalist)
- Generic payment error (Error Recovery)
- No FAQ for prescription upload (Help & Documentation)

14. Advantages

- Quick and cost-effective.
- Can be applied early (sketches, wireframes, prototypes, applications).
- Systematic and structured.
- Identifies common problems.

15. Limitations

- Depends on evaluator expertise.
- Not real user behavior.
- May have false positives.
- Domain knowledge is important.
- Does not replace user testing.

16. Heuristic Evaluation in UX Process

User Research → Requirements → Information Architecture → Heuristic Evaluation → Usability Testing → Redesign → Prototype → Wireframes → Identify Violations → Final Design

Key Takeaways

- Expert-based inspection method.
- Uses Nielsen's 10 heuristics.
- Can be applied to various interfaces.
- Problems are documented and given severity.
- Useful for early detection.
- Does not replace usability testing.
- Combining both gives better results.
- Goal: clear, efficient, consistent, error-tolerant and easy to use.