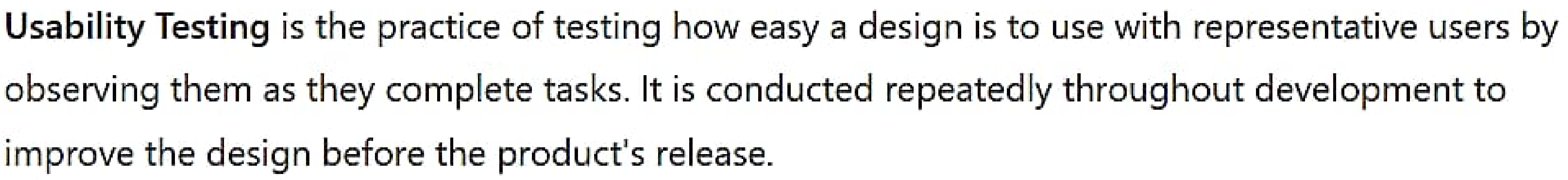


Usability Testing is the practice of testing how easy a design is to use with representative users by observing them as they complete tasks. It is conducted repeatedly throughout development to improve the design before the product's release.

1. **Guerilla Testing**

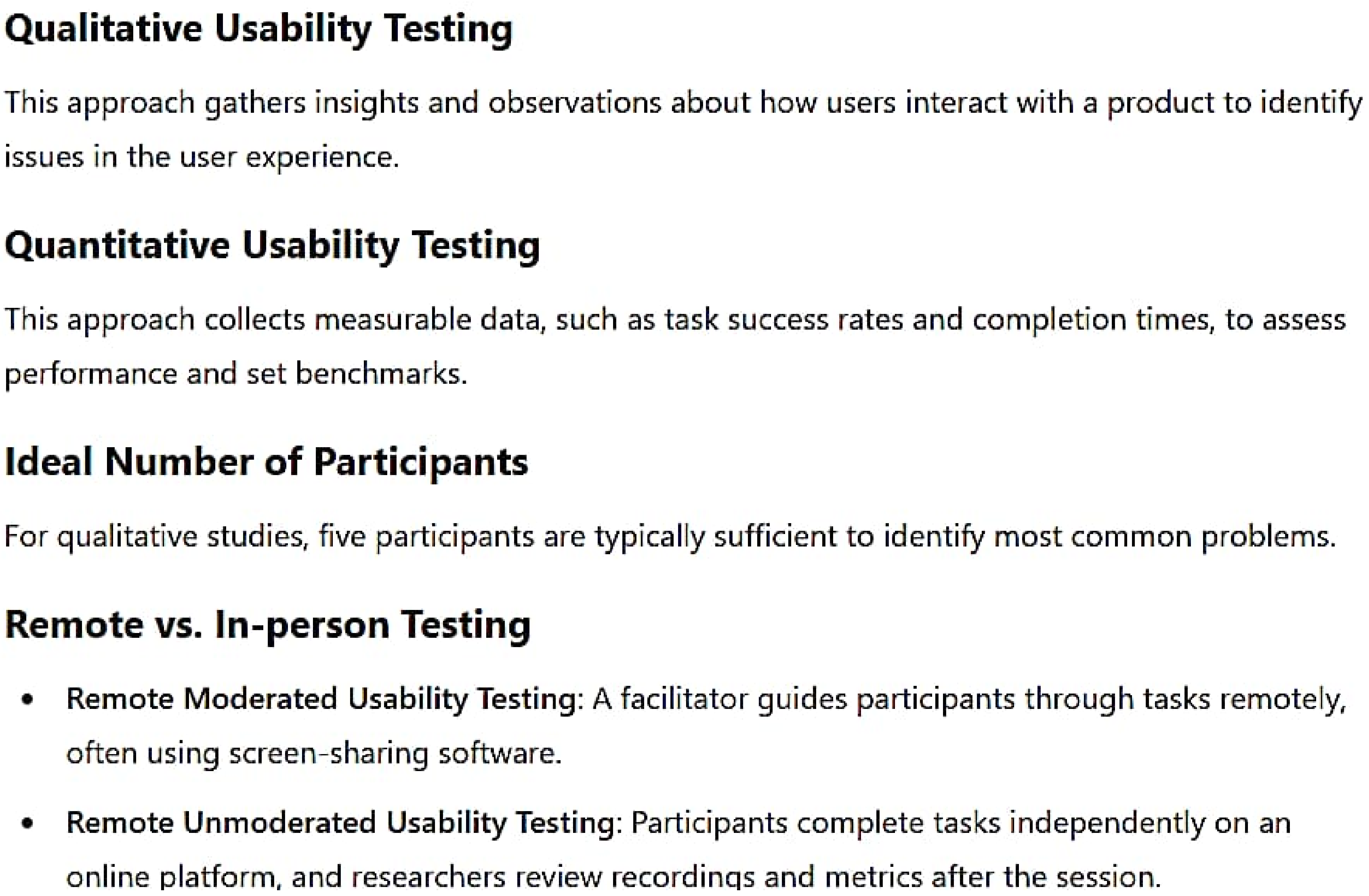
   It is a type of testing where testers ask random users in public places for feedback on a prototype, usually for 5–10 minutes. It is cost-effective, provides instant feedback, and is ideal during the early product development phases.

2. **Usability Lab**

   Conducted in a lab with hired participants, this testing involves moderators guiding 8-10 participants to provide feedback, typically using a tablet or desktop. It is more structured but costlier than guerrilla testing.

3. **Screen or Video Recording**

   This testing records user actions on a product, showing how they navigate and use it. It helps identify usability issues and usually involves 10 users for about 15 minutes.



Qualitative Usability Testing

This approach gathers insights and observations about how users interact with a product to identify issues in the user experience.

Quantitative Usability Testing

This approach collects measurable data, such as task success rates and completion times, to assess performance and set benchmarks.

Ideal Number of Participants

For qualitative studies, five participants are typically sufficient to identify most common problems.

Remote vs. In-person Testing

- **Remote Moderated Usability Testing**: A facilitator guides participants through tasks remotely, often using screen-sharing software.
- **Remote Unmoderated Usability Testing**: Participants complete tasks independently on an online platform, and researchers review recordings and metrics after the session.

Lab Environment

This setting is perfect for early design stages. In a lab, everything can be controlled, making it easier to test low-fidelity prototypes and quickly improve them.

Natural Environment

- **For Consumer Products**: Testing in someone's home shows how real people use the product in their daily lives.
- **For Enterprise Products**: Observing users at their workplace helps you understand how they interact with the product in their professional setting.
- **For Real-World Use**: Testing in natural environments gives you valuable insights into how people actually engage with the product in their normal routines.

Remote Environment

- **Flexibility**: Remote testing lets people participate from anywhere, making it super convenient.
- **Global Access**: You can include users from around the world to get diverse feedback.
- **Realistic Simulations**: It's easy to test how the product works across different screen sizes, operating systems, browsers, and internet speeds.

Essential Tools and Software for Usability Testing

1. **UserTesting**

   A platform for gathering real-time feedback from users to improve the product's usability.

2. **Lookback**

   A remote testing tool with features for both moderated and unmoderated tests, as well as user interviews.

3. **Maze**

   A quick and intuitive tool for testing prototypes, gathering user insights, and validating designs remotely.

4. **Hotjar**

   Tracks user behavior with clicks and mouse movement recordings while also collecting feedback through surveys and forms.

5. **Userlytics Corporation**

   A usability testing tool that supports both moderated and unmoderated tests to enhance user experience.

6. **Optimal Workshop**

   Offers tools for information architecture testing, including tree testing and card sorting, to improve website navigation.

7. **UserZoom**

   Provides moderated and unmoderated testing to evaluate and improve a website's user experience (UX).

8. **Loop11**

   Specializes in unmoderated remote testing with advanced features like A/B testing, heatmaps, and clickstream analysis.

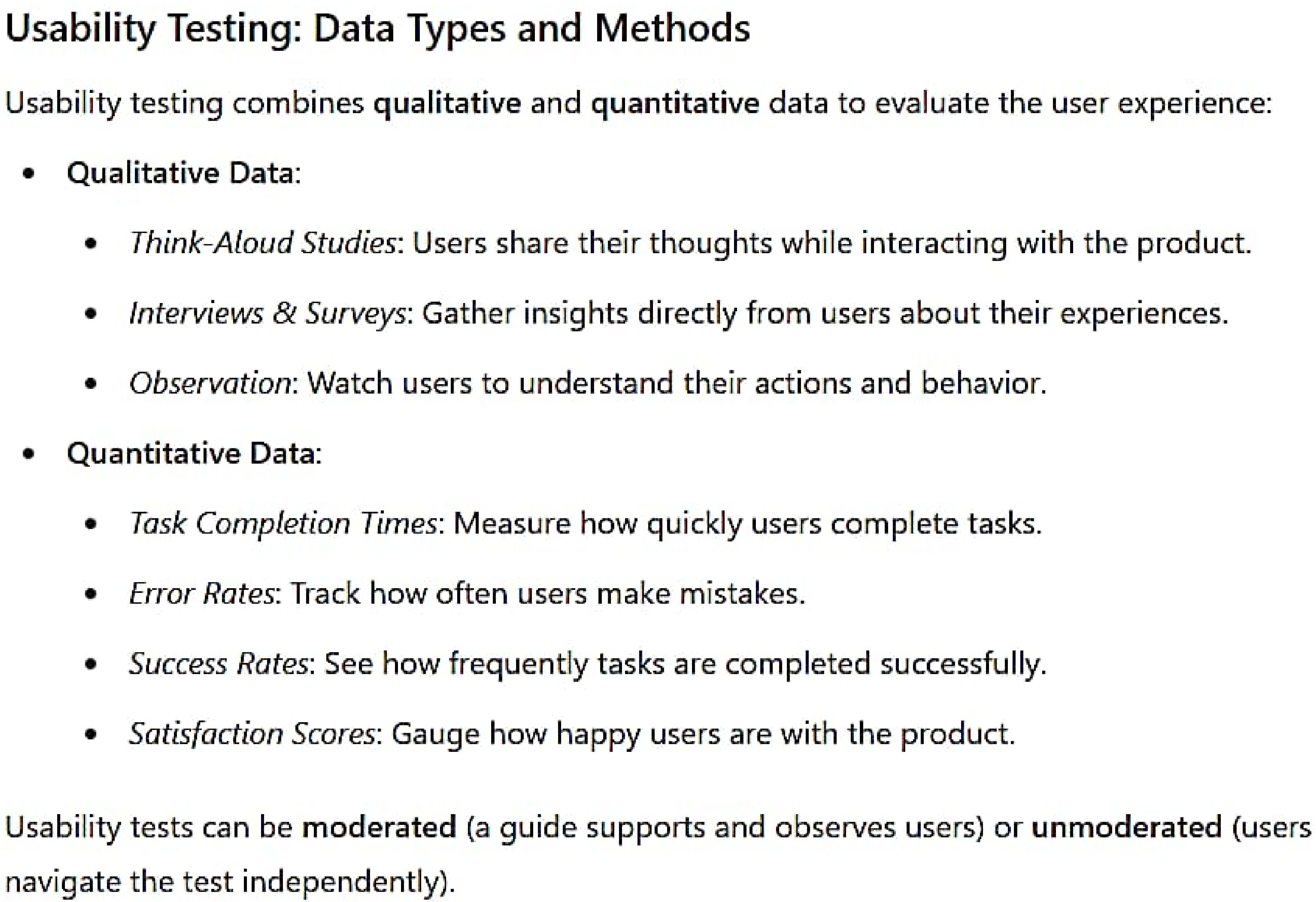
9. **Crazy Egg**

   Helps identify areas on a website that need improvement using tools like heatmaps, scroll maps, and A/B testing.

The primary purpose of a usability test is to gather the data needed to identify usability issues and improve a website's or app’s design.

Objectives of Usability Testing

- **User Satisfaction**: Learn how users feel about the product and make changes to enhance their experience.
- **User Needs**: Identify what users truly need and adjust the product to meet those expectations.
- **User Flows**: Observe how users navigate the product and refine the flow to make it smoother.
- **Content**: See how users interact with the content and improve it to better serve their needs.
- **Ease of Use**: Assess how simple it is for users to complete tasks and make the product easier to use.

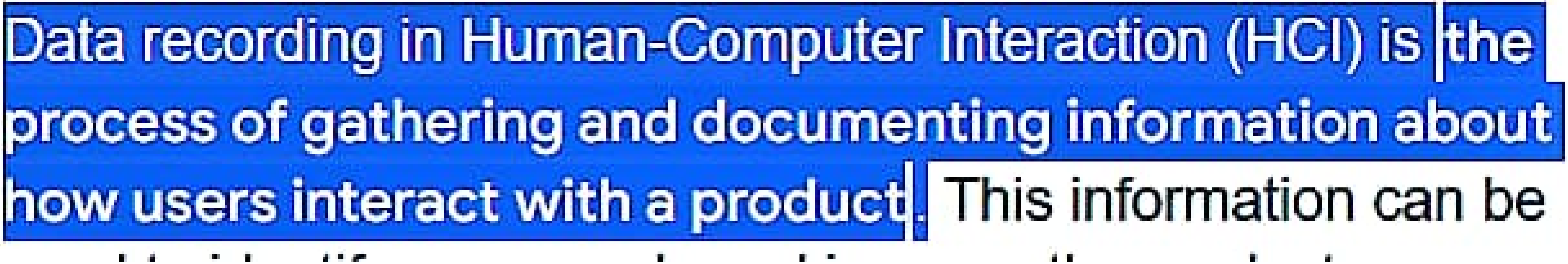


Usability Testing: Data Types and Methods

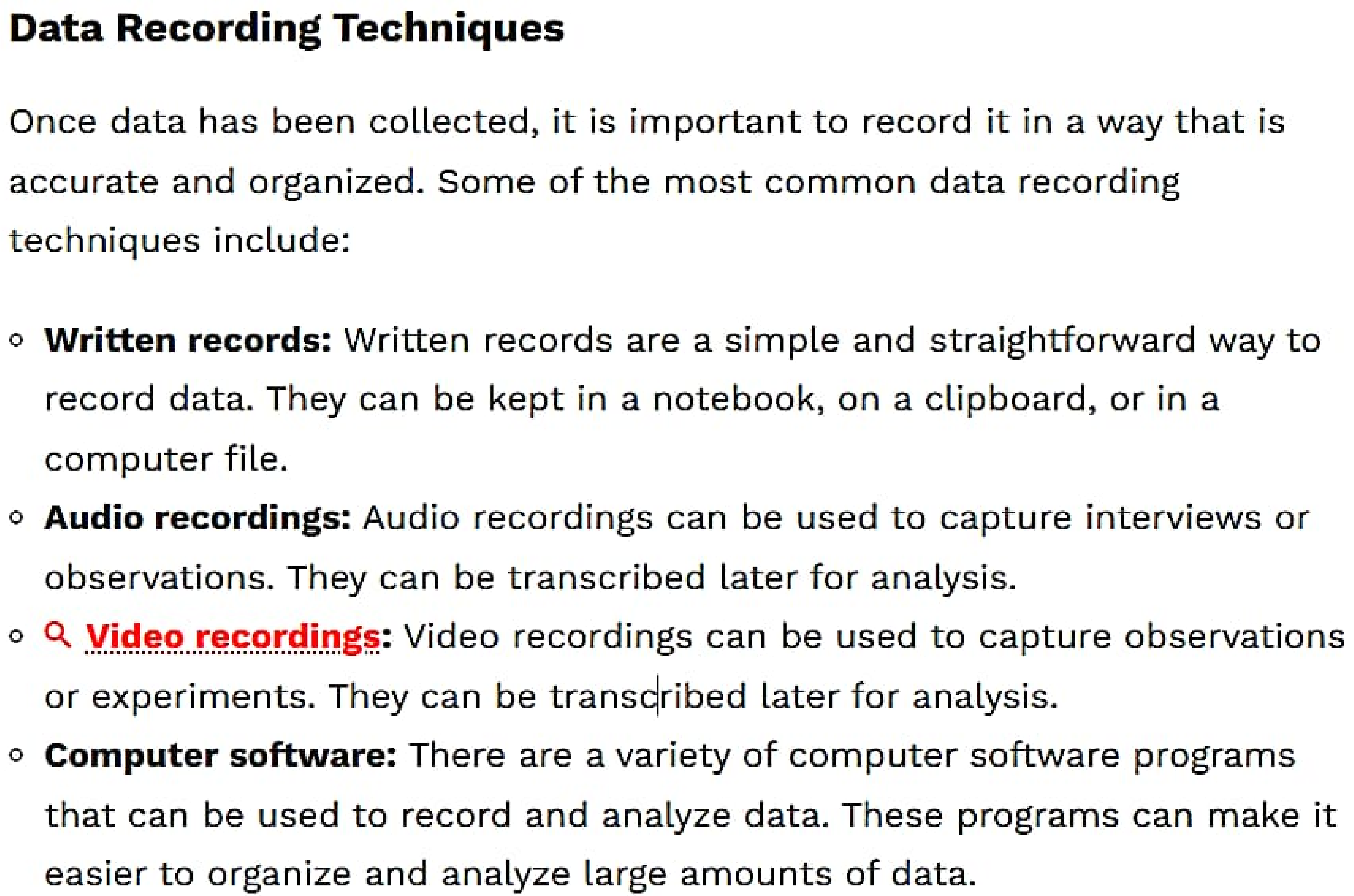
Usability testing combines **qualitative** and **quantitative** data to evaluate the user experience:

- **Qualitative Data**:
    - *Think-Aloud Studies*: Users share their thoughts while interacting with the product.
    - *Interviews & Surveys*: Gather insights directly from users about their experiences.
    - *Observation*: Watch users to understand their actions and behavior.
- **Quantitative Data**:
    - *Task Completion Times*: Measure how quickly users complete tasks.
    - *Error Rates*: Track how often users make mistakes.
    - *Success Rates*: See how frequently tasks are completed successfully.
    - *Satisfaction Scores*: Gauge how happy users are with the product.

Usability tests can be **moderated** (a guide supports and observes users) or **unmoderated** (users navigate the test independently).



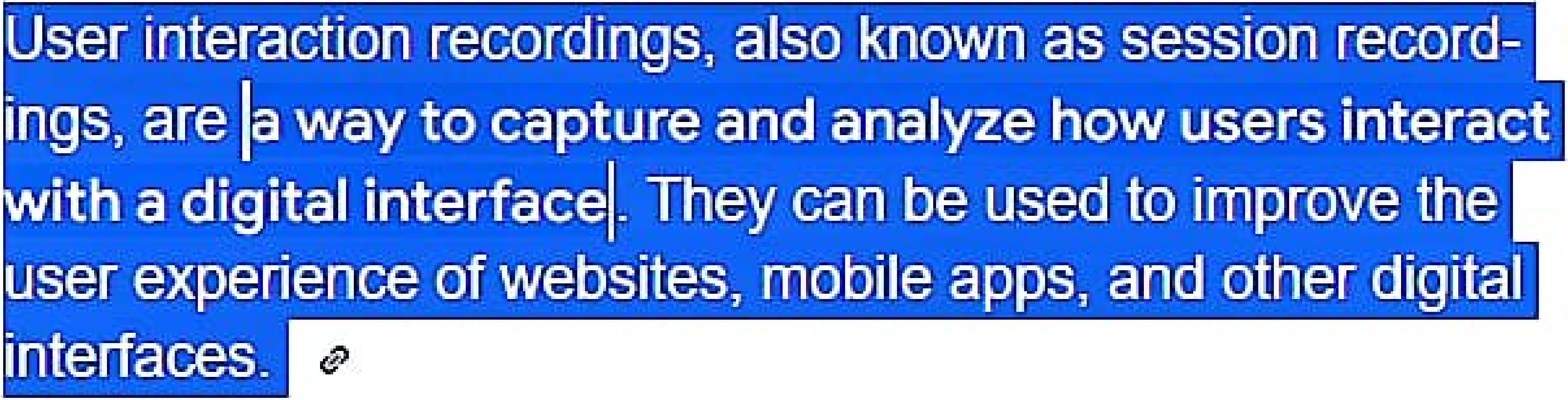
Data recording in Human-Computer Interaction (HCI) is the process of gathering and documenting information about how users interact with a product. This information can be



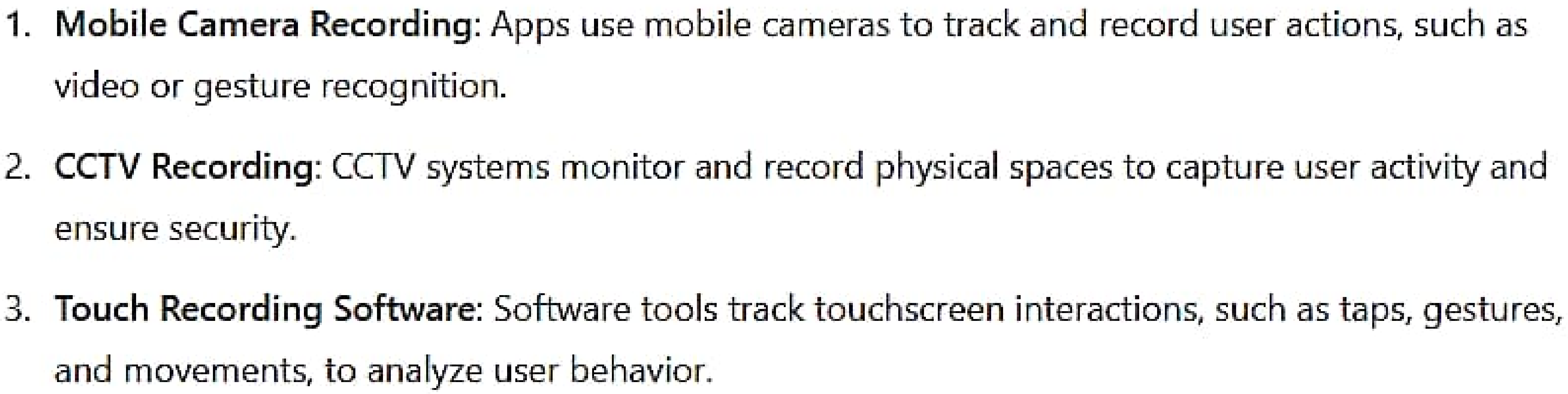
Data Recording Techniques

Once data has been collected, it is important to record it in a way that is accurate and organized. Some of the most common data recording techniques include:

- **Written records:** Written records are a simple and straightforward way to record data. They can be kept in a notebook, on a clipboard, or in a computer file.
- **Audio recordings:** Audio recordings can be used to capture interviews or observations. They can be transcribed later for analysis.
- **Video recordings:** Video recordings can be used to capture observations or experiments. They can be transcribed later for analysis.
- **Computer software:** There are a variety of computer software programs that can be used to record and analyze data. These programs can make it easier to organize and analyze large amounts of data.



User interaction recordings, also known as session recordings, are a way to capture and analyze how users interact with a digital interface. They can be used to improve the user experience of websites, mobile apps, and other digital interfaces.



1. **Mobile Camera Recording:** Apps use mobile cameras to track and record user actions, such as video or gesture recognition.
2. **CCTV Recording:** CCTV systems monitor and record physical spaces to capture user activity and ensure security.
3. **Touch Recording Software:** Software tools track touchscreen interactions, such as taps, gestures, and movements, to analyze user behavior.

Donald Norman's Design Principles of Usability

These principles focus on making systems intuitive, easy to use, and user-friendly by aligning design with human behavior.

---

1. Visibility: "Can I see it?"

- **Concept:** Functions should be visible so users know what to do next. Hidden features make interaction harder.
- **importance:** Helps users understand what works and how to interact with the system.
- **Example:** A flat plate on a door suggests "push," while a vertical handle suggests "pull."

---

2. Feedback: "What is it doing now?"

- **Concept:** The system should communicate what action has been performed using sound, visuals, or animations.
- **Importance:** Keeps users informed and satisfied.
- **Example:** A "click" sound when pressing a button or a red highlight when an error occurs.

3. Constraints: "What can't I do?"

- **Concept:** Limit users to valid actions to prevent errors.
- **Importance:** Guides users to correct usage and reduces mistakes.
- **Types:**
  - **Physical:** Prevent incorrect use (e.g., SIM cards with notched corners).
  - **Logical:** Use logic (e.g., color coding or symbols for actions).
  - **Cultural:** Rely on learned conventions (e.g., trash icons for delete).

---

4. Consistency

- **Concept:** Design similar actions and elements throughout the system.
- **Importance:** Makes the system easy to learn and use while reducing errors.
- **Types:**
  - **Internal:** Consistent operations within an app (e.g., Ctrl+C for copy, Ctrl+S for save).
  - **External:** Consistency across systems (e.g., undo/redo in Word and Google Docs).

5. Mapping: "How does it work?"

- **Concept:** The controls should logically correspond to their effects.
- **Example:** Arrows on a keyboard for moving a cursor up or down.

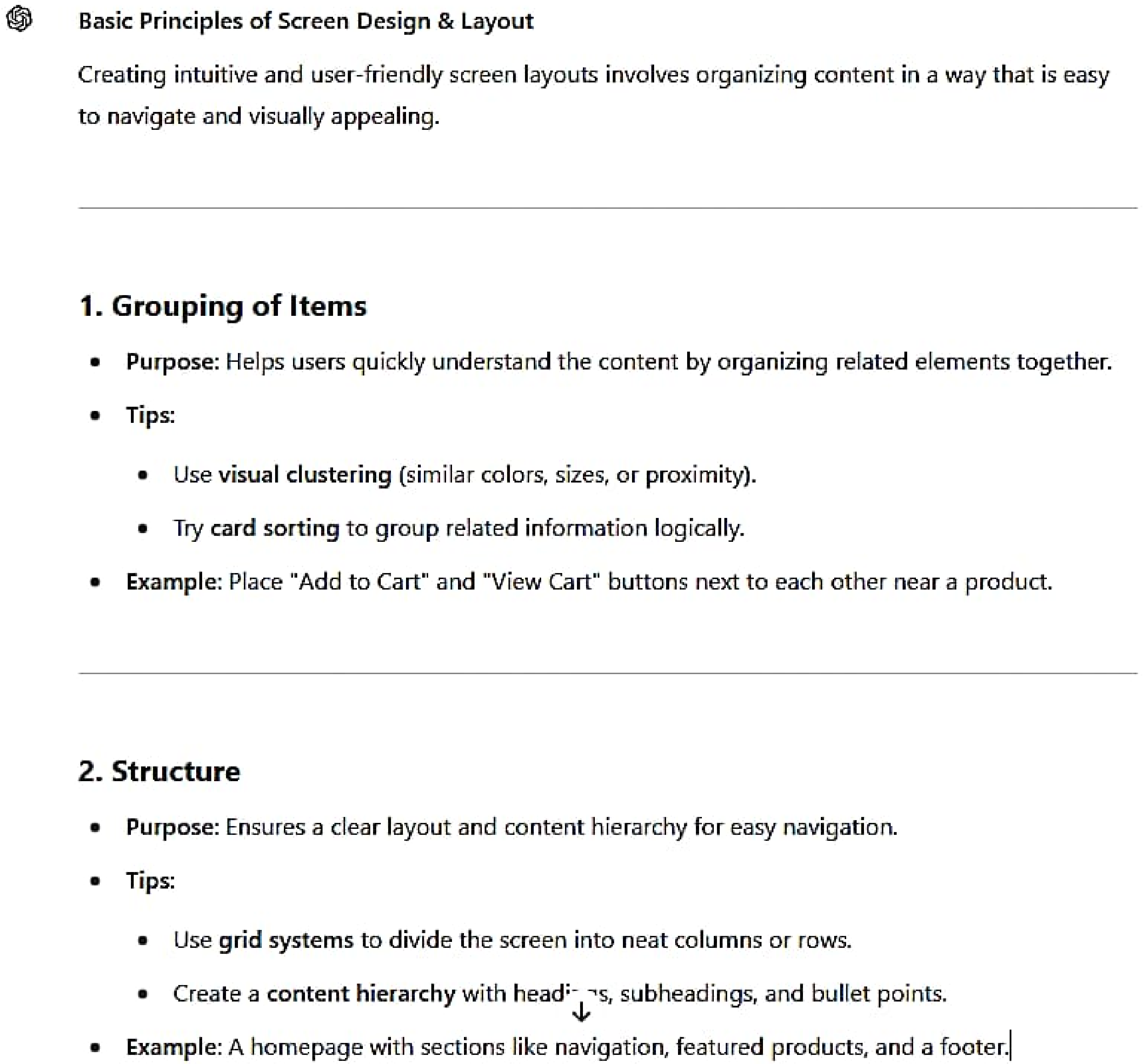
---

6. Affordance: "How do I use it?"

- **Concept:** The design of an object should suggest how it's used.
- **Types:**
  - **Physical:** A door handle implies pulling; knobs imply turning.
  - **Virtual:** A trash icon implies deletion; a scroll bar implies scrolling.

Interconnection Between Principles

- **Visibility** informs **affordance** and **feedback.**
- **Feedback** reinforces **affordance** and **mapping**.
- **Constraints** improve **affordance** and **consistency.**
- **Mapping** depends on **visibility** and **consistency.**



Basic Principles of Screen Design & Layout

Creating intuitive and user-friendly screen layouts involves organizing content in a way that is easy to navigate and visually appealing.

---

1. Grouping of Items

- **Purpose:** Helps users quickly understand the content by organizing related elements together.
- **Tips:**
  - Use **visual clustering** (similar colors, sizes, or proximity).
  - Try **card sorting** to group related information logically.
- **Example:** Place "Add to Cart" and "View Cart" buttons next to each other near a product.

---

2. Structure

- **Purpose:** Ensures a clear layout and content hierarchy for easy navigation.
- **Tips:**
  - Use **grid systems** to divide the screen into neat columns or rows.
  - Create a **content hierarchy** with headings, subheadings, and bullet points.
- **Example:** A homepage with sections like navigation, featured products, and a footer.

3. Order of Items

- **Purpose:** Guides users by presenting content in a logical flow, prioritizing key elements.
- **Tips:**
  - Design for **Z-pattern** scanning (place CTAs along the Z-path).
  - Leverage the **F-pattern** for important elements in the top-left or left column.
- **Example:** A form with key fields like "Name" and "Email" at the top and less critical ones later.

---

4. Alignment

- **Purpose:** Creates a clean, organized look by ensuring elements are properly lined up.
- **Tips:**
  - Use **edge alignment** (left, right, or center) for consistency.
  - Ensure even **vertical and horizontal spacing** for balance.
- **Example:** A perfectly left-aligned navigation menu for a structured appearance.

---

5. Whitespace

- **Purpose:** Improves readability and reduces visual clutter by using empty space effectively.
- **Tips:**
  - Add **padding** around buttons or text to avoid crowding.
  - Use **margins** to separate sections and create breathing room.
- **Example:** A webpage with large margins around text and buttons for a clean, distraction-free design.

USABILITY TASK ANALYSIS IN HCI:

Usability Task Analysis in HCI is the process of studying the tasks users perform when interacting with a system. It helps identify difficulties, improve the design, and ensure that the system supports users in achieving their goals efficiently.

Hierarchical Task Analysis (HTA):

HTA is a method used to break down tasks into a hierarchy of sub-tasks. It focuses on understanding how users perform a task by dividing it into main tasks and smaller, detailed actions.

How it works:

- **Top-level tasks**: Broad actions or goals the user wants to achieve (e.g., "Complete a purchase").
- **Sub-tasks**: These are smaller actions that help complete the top-level task (e.g., "Add items to cart," "Enter shipping details").
- **Sub-sub-tasks**: Even smaller actions that are part of the sub-tasks (e.g., "Select product," "Click 'Add to Cart'").

Example:

Main Task: "Complete an online purchase"

- **Sub-task 1**: "Browse products"
  - **Sub-sub-task 1**: "Search for a product"
  - **Sub-sub-task 2**: "Filter by category"

- **Sub-task 2**: "Add to cart"
    - **Sub-sub-task 1**: "Select quantity"
    - **Sub-sub-task 2**: "Click 'Add to Cart'"
- **Sub-task 3**: "Enter payment details"
    - **Sub-sub-task 1**: "Enter credit card number"
    - **Sub-sub-task 2**: "Submit payment"

Why it's important:

- Helps designers understand the user's journey and potential pain points.
- Breaks down complex tasks into simpler, manageable actions.
- Ensures that the interface supports users in completing tasks smoothly.