Addis Software

Designing for Developers: Bridging UI/UX and Code

Miheret Merid

20/12/2024

Outline

1. Objective

- Equip developers with essential UI/UX design principles to improve collaboration

- Foster empathy for users, and

- Enable them to contribute to design discussions and processes.
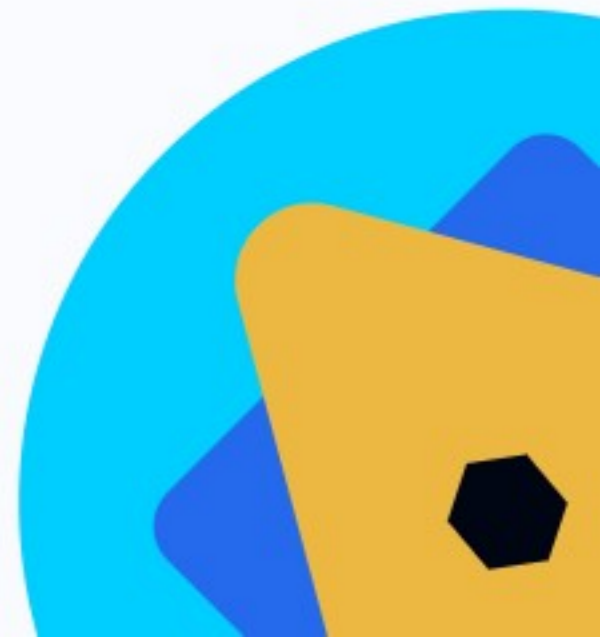
2. Introduction to UI/UX Design
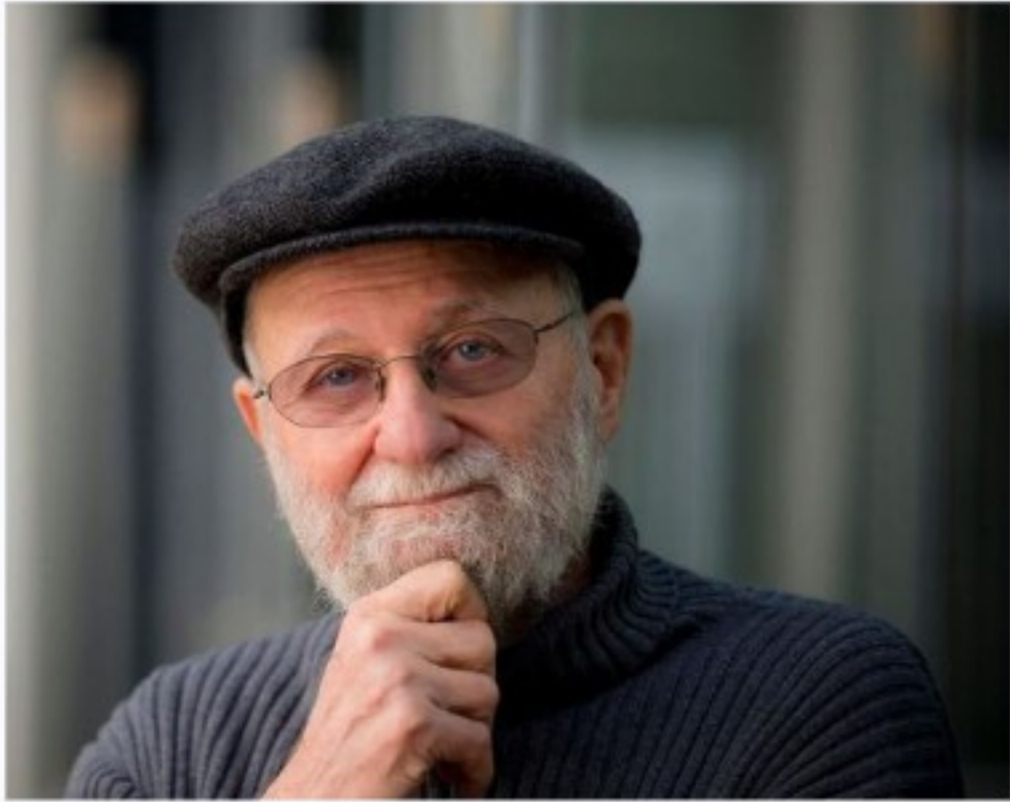
UX (User Experience)

- Focuses on the overall experience of using a product or system.

UI (User Interface)

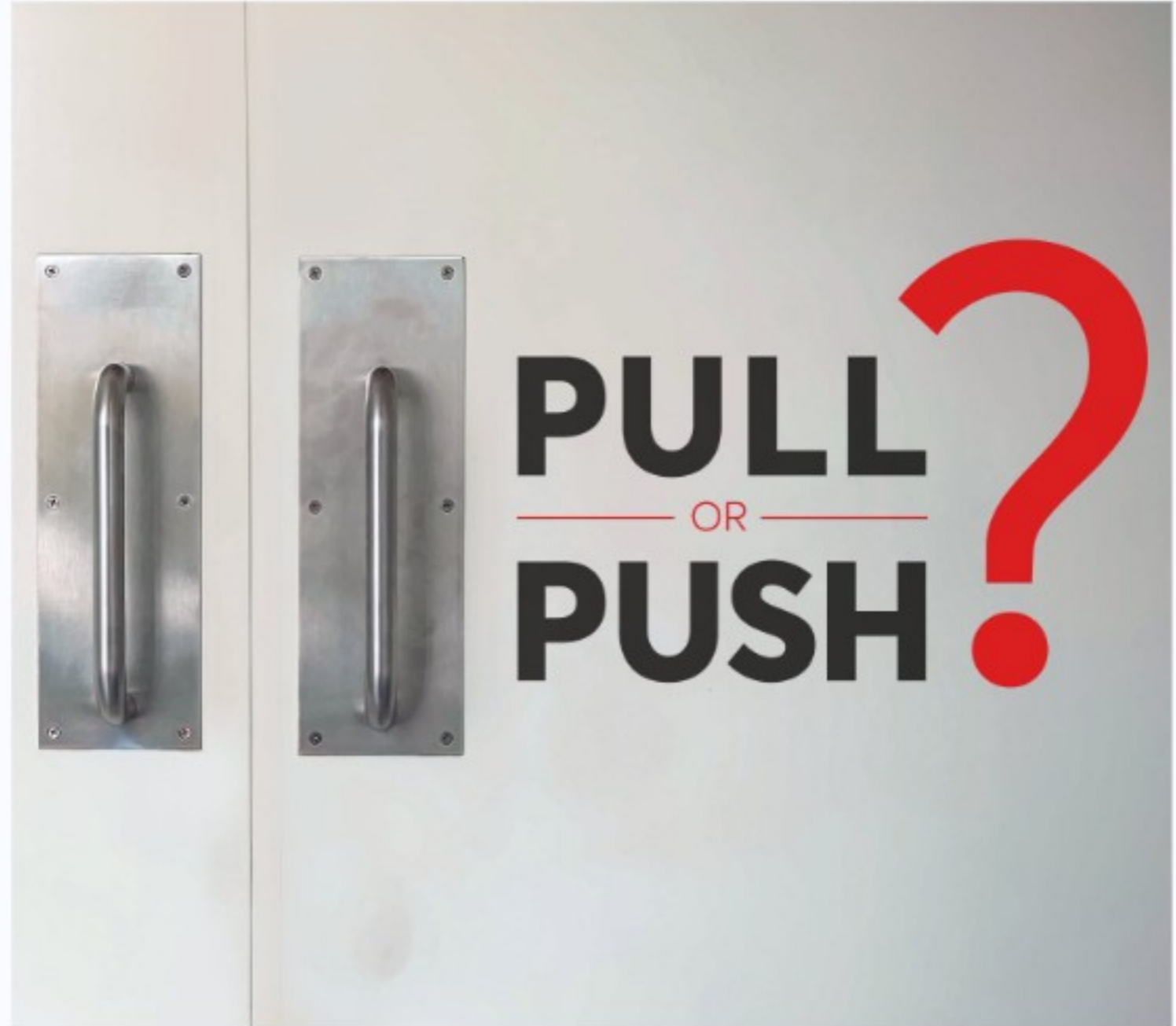- Focuses on the visual(aesthetics, layout, and functionality of the interface.

In essence, UX is about how it works, while UI is about how it looks.
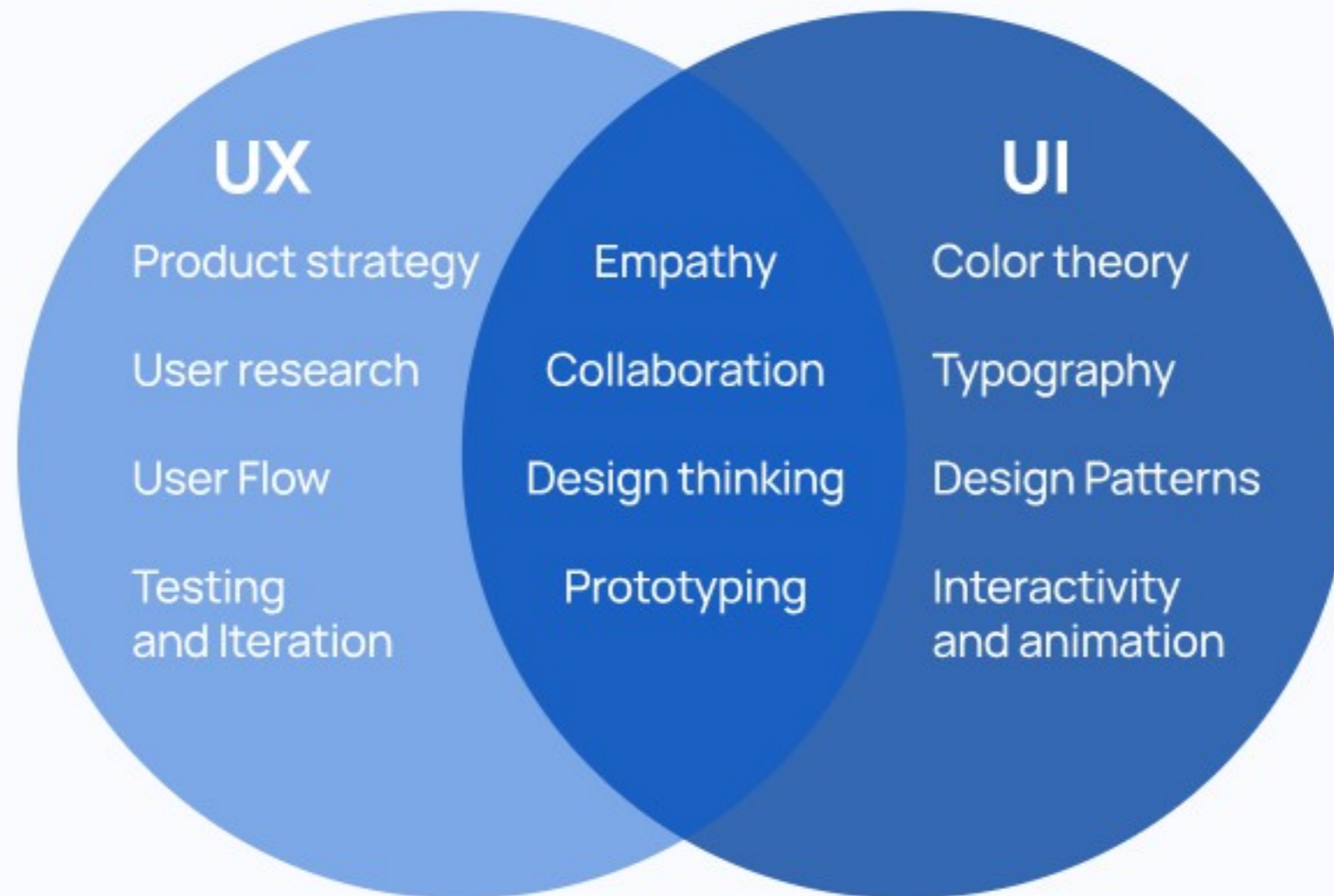

User experience

Design

Donald A. Norman (Father of UX)

"I push doors that are meant to be pulled, pull doors that should be pushed, and walk into doors that neither pull nor push, but slide."

PULL
OR
PUSH?

3. Difference between UI vs. UX?

UX
- Product strategy
- User research
- User Flow
- Testing and Iteration

Overlap:
- Empathy
- Collaboration
- Design thinking
- Prototyping

UI
- Color theory
- Typography
- Design Patterns
- Interactivity and animation
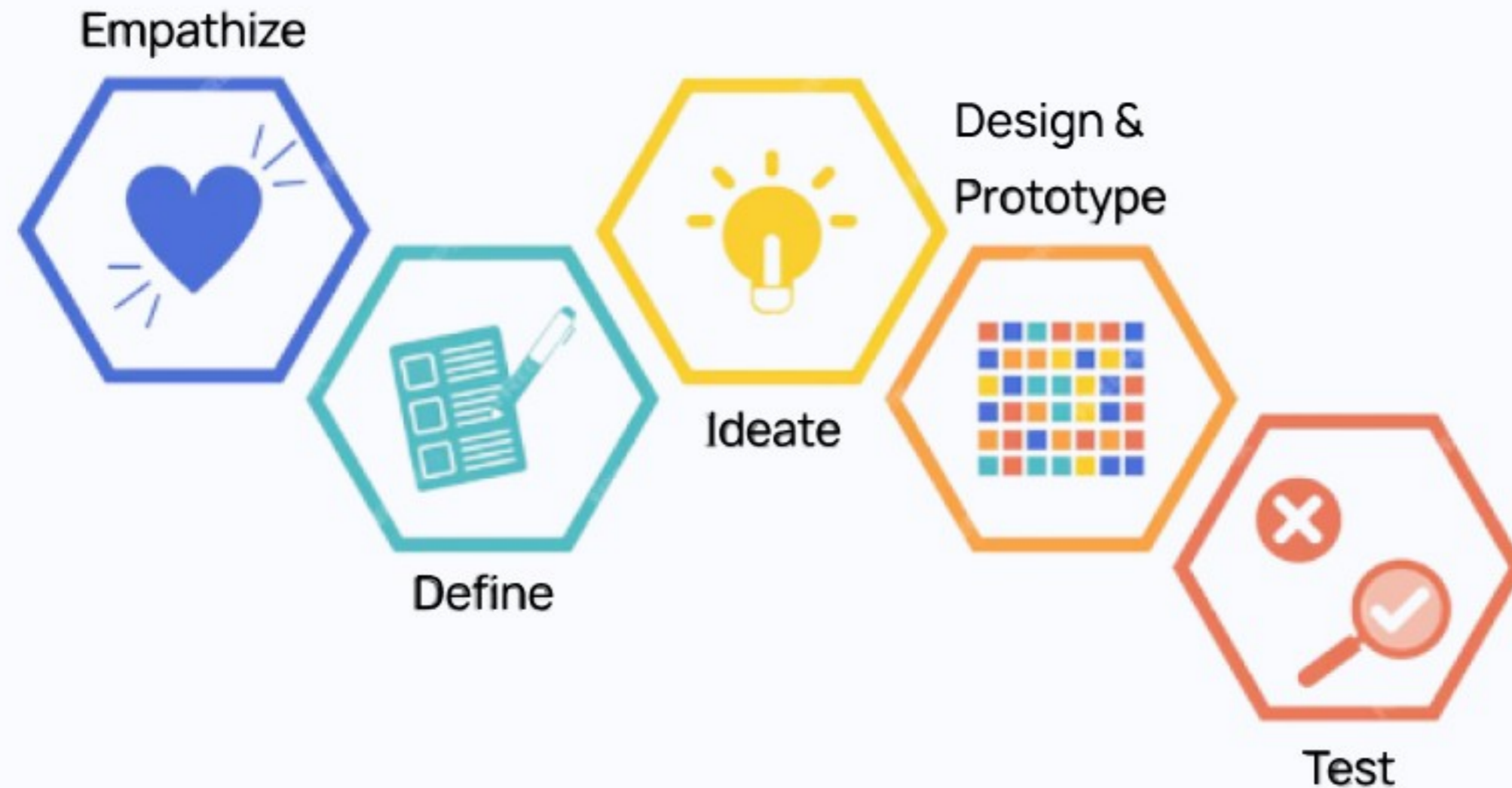
3.1 Color Theory

Orange

Blue

Red

Purple

Green

Black

3.2 Design Thinking

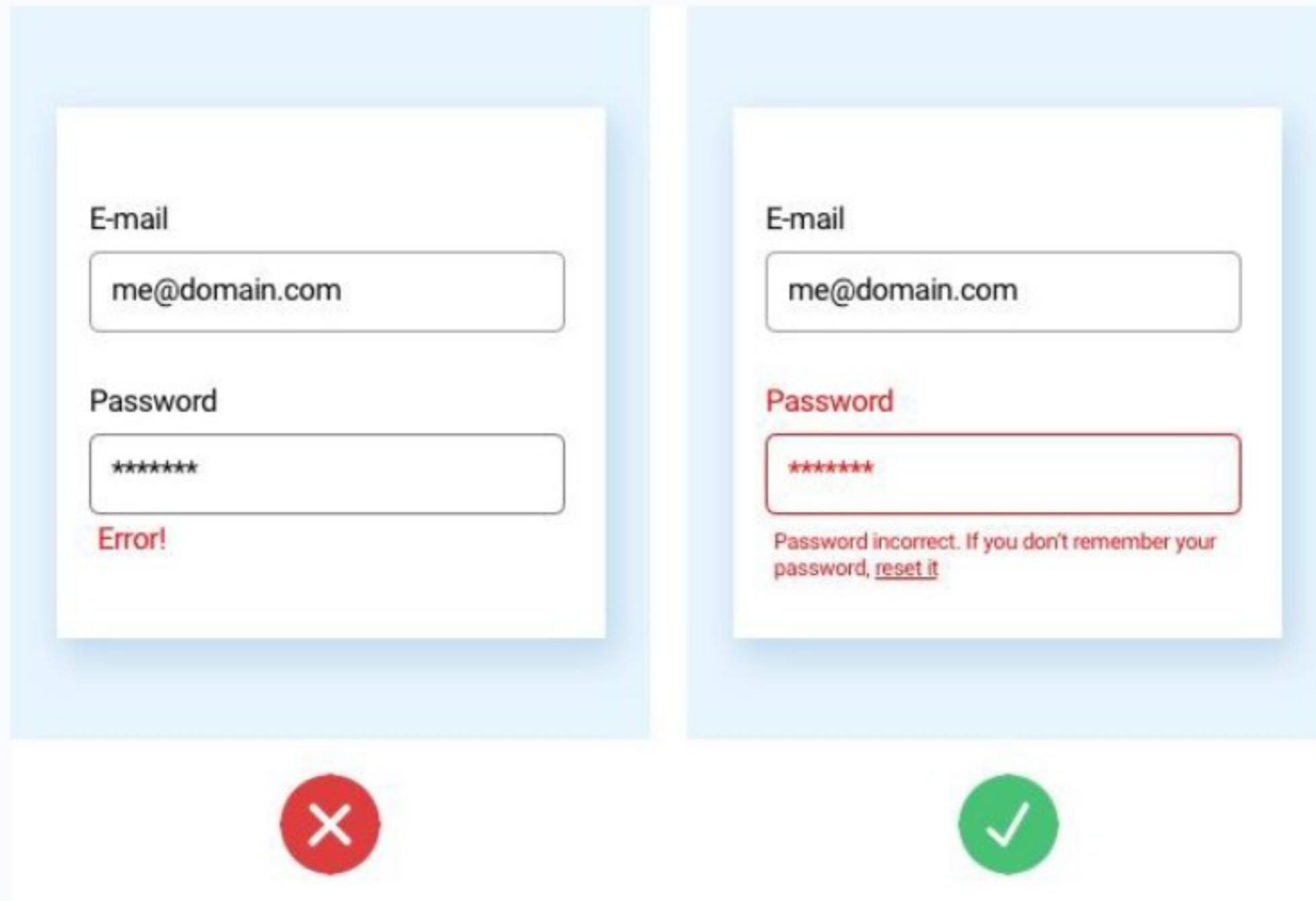Design thinking is concerned with solving problems through design.

Good UX

- Intuitive and Easy to Use
- Efficient and Time-Saving
- Accessible and Inclusive
- Consistent and Predictable
- Engaging and Enjoyable

Bad UX

- Confusing and Unintuitive
- Time-Consuming
- Poor readability (Inaccessible)
- Inconsistent Design
- Poor error handling

An image comparing two forms: on the left, poor error handling with vague or unhelpful messages; on the right, effective error handling that provides clear, informative feedback to guide users toward correction.

4. Importance of good UX

- **Reduces Support Tickets:** (e.g, clear labels, tooltips, and onboarding can resolve 80% of common user issues)

- **Improves User Retention:** (e.g, Netflix's recommendation algorithm, driven by UX principles, encourages repeat engagement by offering relevant content to each user.)

- **Makes Code More Maintainable** (Consistent Design System: UI elements (buttons, fonts, colors), developers can reuse components across projects, reducing the complexity of the code and simplifying future updates. This consistency minimizes errors and optimizes the codebase)

5. The Developer's Role in UI/UX

5.1 Overlap Between Design and Development

UI/UX design and development are closely intertwined. The overlap occurs when designers provide the blueprints—such as wireframes, user flows, and prototypes—and developers work to bring them to life with code. The key areas of overlap include:

- **Responsive Design:** Developers must implement responsive CSS techniques, while designers provide mockups that outline the necessary breakpoints.
- **Consistency and Visual Language:** Developers need to understand the design language (colors, typography, icons) to ensure that the final product aligns with the intended design.
- **Prototyping and Testing**: Designers create interactive prototypes to test the user journey, but developers may assist in creating functional prototypes, enhancing feedback loops and testing accuracy.

5.2 How Understanding Design Leads to Smoother Workflows

- **Efficient Implementation**: Speeds up accurate development with fewer revisions.
- **Reduced Design Overhaul**: Identifies technical issues early, avoiding major changes later.
- **Better Collaboration**: Aligns development choices with user-focused design goals.

This mutual understanding ensures cohesive workflows, user-friendly products, and high-quality results.

5.3 Collaboration Best Practices

Reading Design Specs in tools like Figma **Guide to Inspecting**
What to look for: Spacing, sizes, colors, and interactions.

6. The Future of Design and Developers' Role

1. AI-Powered Personalization

Personalization uses AI algorithms to analyze user behavior and preferences, delivering tailored experiences.

Examples: Netflix recommending shows, Spotify curating playlists, or e-commerce sites suggesting products.

How It Impacts Design:

- Interfaces become dynamic, adapting to individual users in real-time.
- Designers and developers need to consider flexible layouts and modular components to support this adaptability.

Netflix recommending shows

6. The Future of Design and Developers' Role

1. AI-Powered Personalization

Developer's Role:

- **Integrating AI Models:** Developers implement APIs or libraries (e.g., TensorFlow.js or OpenAI APIs) to enhance personalization.

- **Optimizing Performance:** Ensure fast processing of AI-driven recommendations, even on resource-constrained devices.

- **Data Security:** Protect user data powering these algorithms by following best practices for encryption and compliance.

6. The Future of Design and Developers' Role

2. Zero UI

Moving beyond screens to gesture-based, voice-controlled, or presence-detection interfaces.

Examples: Smart speakers like Alexa, motion-controlled gaming, or wearables like smart glasses.

How It Impacts Design:

- Shifts focus from visual design to natural interaction patterns.
- Prioritizes usability in environments with no screens or minimal UI.

Alexa

Motion-controlled gaming

Smart glasses

6. The Future of Design and Developers' Role

2. Zero UI

Developer's Role:

- **Implementing Sensors & APIs:** Use technologies like speech recognition, motion tracking, or IoT frameworks.
- **Prototyping Future Interfaces:** Collaborate with designers to create intuitive gestures and commands.
- **Testing for Usability:** Ensure seamless interaction and low latency in gesture or voice responses.

Thank You

Designing for Developers: Bridging UI/UX and Code

miheret@addissoftware.com