HUMAN COMPUTER INTERACTION (M23DES212)

School of CSA /MCA- IInd sem

Dr. Ambili P S

UNIT 4

Unit Title	Design and Development
Employing User Personas	
Prototyping Tools and Techniques	
Scenario-Based Design, Participatory Design	
Application-/Domain-Specific Design: Health Care, Aerospace, kids	

USER PERSONAS

USER PERSONAS

Personas Move Us Beyond the Word "User"

1. Personas are fictitious, specific, concrete representations of target users.
2. Personas put a face on the user; a memorable, engaging, and actionable image to serve as a design target.
3. Personas add the detail and specificity needed to provide product development teams the understanding needed to create user centered products.
4. Personas humanize vast and disparate data sources, capitalize on our ability to remember details about individual people, and, in so doing, provide a usable and useful alternative to referring to "the user."
5. Personas create a concrete, focused, and stable definition of a specific audience.
6. Personas were born out of a short tradition in the User- Centered Design (UCD) community toward user and system modeling and out of a somewhat longer tradition in marketing around market definition and customer representation.

USER CENTRIC DESIGN

- User-centered design is a concept that aims to prioritize the needs of all users in product development.

It is challenging to achieve this due to several factors.

1. Firstly, being user-centered is not natural, as our natural tendency is to design based on our own wants and needs.
2. Second, users are complex and varied, and understanding their needs, desires, preferences, and behaviors is crucial.
3. Third, those conducting user and market research are not typically the people designing and building the product.

- If every one in the organization doesn't have a clear and consistent understanding of whom they are building the product for, the product can fail.

EXAMPLES: HEALTH CARE DOMAIN

Persona: Dr. Meera Sharma

- **Age**: 38
- **Role**: General Physician at a government hospital
- **Tech Use**: Laptop + smartphone for patient consultations
- **Goals**: Quickly review patient history, conduct online consultations, and prescribe digitally
- **Frustrations**:
  - Patients often don't upload correct health records
  - App interfaces are cluttered with too many options

EXAMPLES: AEROSPACE DOMAIN

Persona: Captain James Miller
Age: 45
Role: Commercial Airline Pilot (15 years experience)
Tech Use: Cockpit displays, flight management systems
Goals: Respond to alerts quickly, maintain flight safety, reduce cognitive overload
Frustrations:

- Too many alerts with similar sounds → hard to prioritize in emergencies
- Critical data sometimes buried in menus

EXAMPLES: DESIGNING AN *EDUCATIONAL GAME APP FOR CHILDREN*

Persona: Aarav

Age: 7

Role: Primary school student

Tech Use: Shared tablet at home, supervised by parents

Goals: Learn math in a fun way, collect rewards/badges, play with friends

Frustrations:

- Gets bored with too much text
- Complex instructions confuse him

Needs: Bright visuals, simple tasks, voice instructions, instant rewards

THE PERSONA LIFECYCLE

- The persona lifecycle is a symbolic framework that breaks down persona creation and use into sequential phases that map on to the life stages of human reproduction and development.

This framework is composed of five phases:

(1)Family Planning

(2) Conception & Gestation

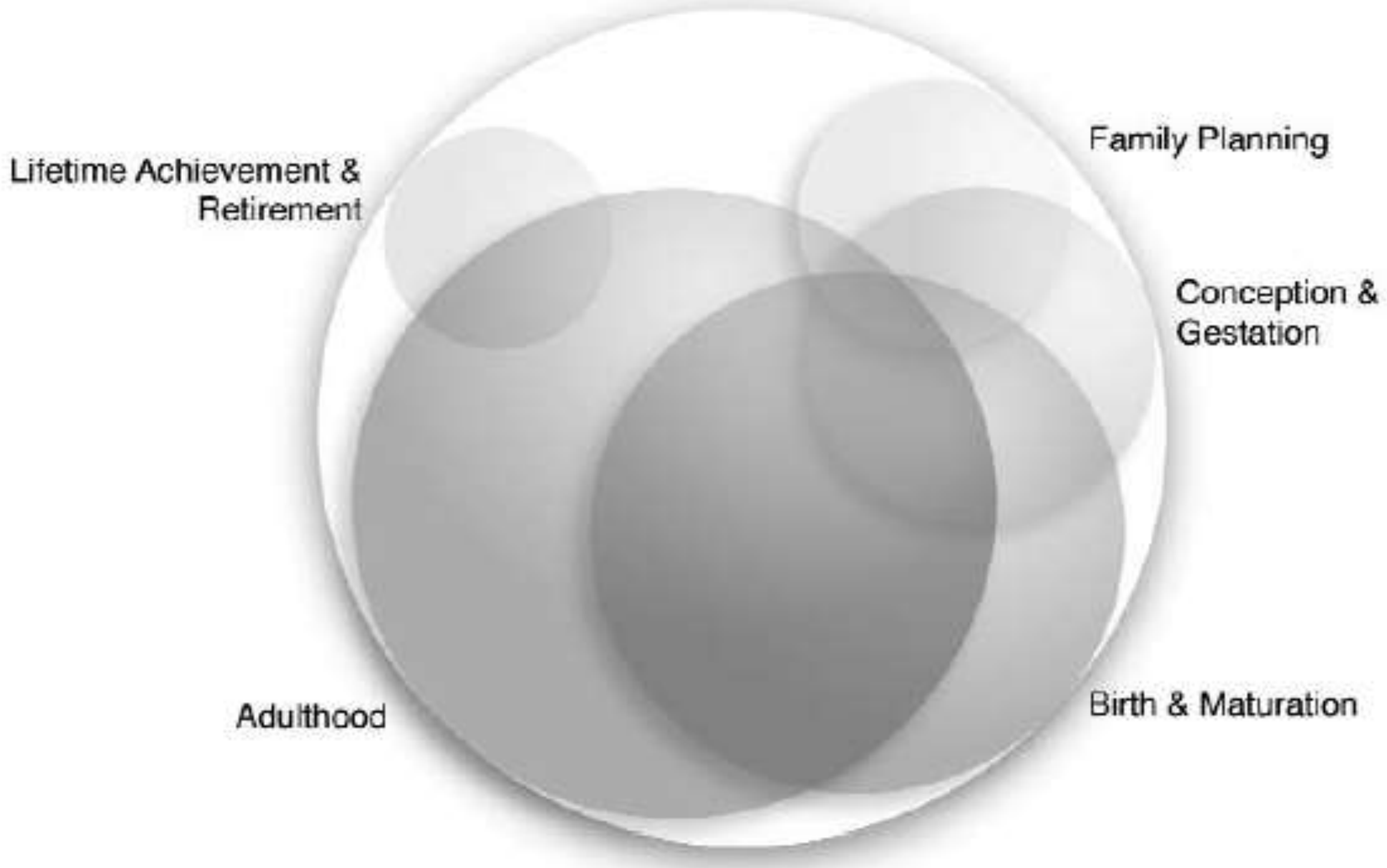
(3) Birth & Maturation

(4) Adulthood

(5) Lifetime Achievement & retirement

THE PERSONA LIFECYCLE

This Persona Lifecycle framework is from user-centered design research, where personas are treated like “living entities” that go through stages, just like humans do.



MAKING PERSONAS PRACTICAL

1. When the first started talking to persona practitioners in 2000, it is noticed that many people in the UCD community could see the value of personas in their own work and to their organizations.
2. Those that tried to create and use personas, however, were running into a somewhat consistent set of problems in their persona efforts.
3. While the idea of creating a set of target users is reasonably straightforward, the actual process to create and use personas can be quite complex.

MAKING PERSONAS PRACTICAL

1. Many of the same questions repeatedly:
    1. How do you decide whether personas are the right thing to do in your organization?
    2. How do you incorporate data into personas? What kinds of data work?
    3. How do you know if personas are worth the effort it takes to create them?
    4. How do you communicate personas once they are created?
    5. How do you use personas to design great products?
    6. What do you do with personas once a project is finished? Can you reuse personas?

MAKING PERSONAS PRACTICAL

1. Those that were able to create datadriven personas were finding that, if they are not well communicated and managed, even well-crafted personas are easy for designers and developers to ignore.
2. At worst, poorly executed persona efforts yield no increase in user-focus and leach time and resources from other UCD techniques and other methods that can improve product quality.

MAKING PERSONAS PRACTICAL

1. It is found four common reasons for the failure of persona efforts:
   1. The effort was not accepted or supported by the leadership team,
   2. The personas were not credible and not associated with methodological rigor and/or data,
   3. The personas, and the techniques for using personas, were not well communicated,
   4. The product design and development team did not understand how to use the personas.
2. Once we fully understood the questions and common causes of failure, it is focused our attention on finding solutions based on the input and insights of dozens of persona practitioners

PROTOTYPING TOOLS AND TECHNIQUES

- A prototype is an early model or mock-up of a product that demonstrates its functionality, layout, or interaction flow.
- A prototype is a tangible artifact, not an abstract description that requires interpretation.

Low-Fidelity Tools	High-Fidelity Tools
Simple, rough, inexpensive tools for quick idea exploration.	Advanced, polished, and interactive tools that closely resemble the final product.
Early brainstorming, exploring concepts, checking basic structure and flow.	Testing usability, visual design, and interactions with stakeholders or end users.
Minimal detail – only layouts, structure, or flow (no colors/fonts).	Very detailed – includes UI design, colors, fonts, animations, clickable elements.

PROTOTYPING TOOLS AND TECHNIQUES

Low-Fidelity Tools	**High-Fidelity Tools**
Low resemblance – looks quite different from the final product.	High resemblance – looks and feels like the final product.
Usually static (paper sketches, simple wireframes).	Highly interactive (clickable prototypes, coded demos).
Balsamiq, Miro, Lucidchart.	Figma, Adobe XD, Sketch, InVision, Axure.
Paper sketches, sticky notes, whiteboard flows, cardboard mockups.	3D printing, Arduino prototypes, hardware mockups.

PROTOTYPES AND THE DESIGN PROCESS: USER-CENTERED DESIGN

1. HCI is a user-centered and iterative field that prioritizes user experience and usability.
2. Prototypes help designers identify functional requirements, usability issues, and performance issues early on.
3. Iterative design involves multiple design-implement-test loops, allowing designers to generate and improve upon ideas.
4. Prototypes reveal strengths and weaknesses, can be contextualized, and can be compared with existing systems.
5. They help designers reanalyze user needs in the context of the system being built, enhancing the overall design process.

PROTOTYPING STRATEGIES

1. Horizontal prototypes
2. Vertical prototypes
3. Task-oriented prototypes
4. Scenario-based prototypes

Horizontal = Broad, shallow (all features, low detail).
Vertical = Narrow, deep (one feature, high detail).
Task-Oriented = Focused on specific critical tasks.
Scenario-Based = Story-driven, realistic usage context.

HORIZONTAL PROTOTYPES

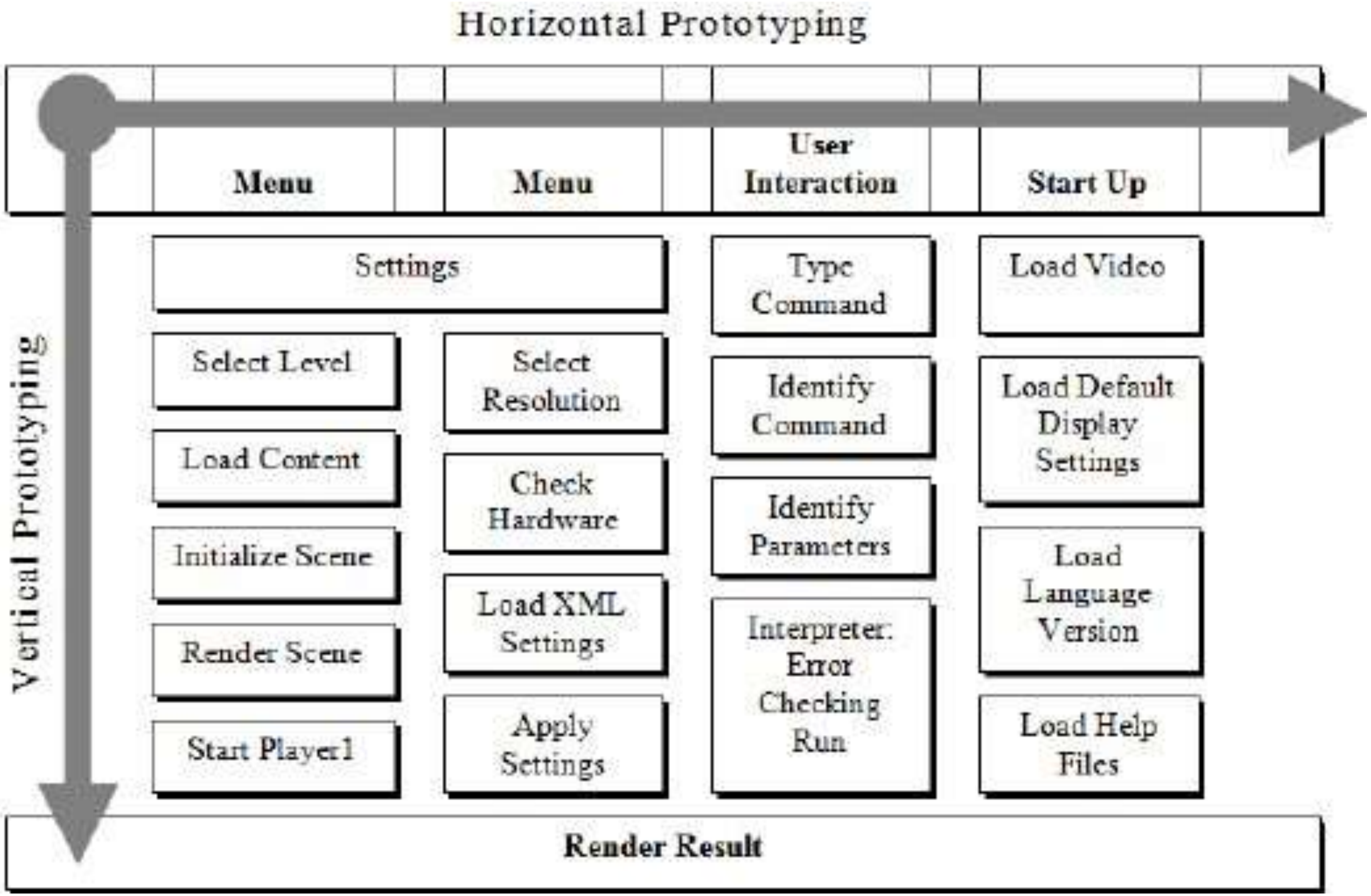
1. Provide a broad view of the system’s features but with little or no functional depth.
2. They are usually used for UI mockups.
3. They help to understand the system from the user's perspective and address issues like consistency, coverage, and redundancy.
4. These prototypes can begin with rapid prototypes and progress to working code.
5. Show the overall navigation, layout, and major options available.

Example: A mobile banking app prototype that shows screens for home, transfer, bill payments, and profile, but the buttons don’t actually execute real transactions.

VERTICAL PROTOTYPES

1. Provide a deep, functional slice of one specific feature while ignoring others.
2. Purpose is to test the technical feasibility and user experience of a key function.
3. They are high-precision prototypes that validate ideas at the system level.
4. They are often created early in the project, before the overall architecture is decided, and focus on one design question.

Example: In the same mobile banking app, the prototype only focuses on the fund transfer feature, showing the step-by-step interaction and even simulating validation and confirmation, while other features are not functional.


Horizontal Prototyping
Vertical Prototyping
Menu
Menu
User Interaction
Start Up
Settings
Select Level
Load Content
Initialize Scene
Render Scene
Start Player1
Select Resolution
Check Hardware
Load XML Settings
Apply Settings
Type Command
Identify Command
Identify Parameters
Interpreter: Error Checking Run
Load Video
Load Default Display Settings
Load Language Version
Load Help Files
Render Result

TASK-ORIENTED PROTOTYPES

1. Task analysis is a crucial process for user interface designers to identify individual tasks that require specific system functionality.
2. Task-based prototypes organize tasks independently, allowing for independent testing.
3. Task-oriented prototypes combine horizontal and vertical prototypes, allowing for detailed analysis of task support.

Example: In an e-commerce website, a prototype is built only for the “Add to Cart → Checkout → Payment” process, without including product reviews, recommendations, or account settings.

SCENARIO-BASED PROTOTYPES

1. Scenario-based prototypes focus on a realistic scenario of a system's use in a real-world setting, avoiding individual tasks.
2. They include both common and unusual situations and explore patterns of activity over time.
3. Anecdotes from real users are used to create use scenarios, which are then transformed into design scenarios.
4. These scenarios are used to create video or software prototypes, allowing users to experience the system in a realistic use context.

SCENARIO-BASED DESIGN

- A user-centered design approach where scenarios (stories of how people use a system in real-life situations) are used to drive requirements, design, and evaluation.
- Scenario-based design changes the focus of design work from defining system operations (e.g., functional specification) to describing how people will use a system to accomplish work tasks and other activities.
- It emphasizes contextual use instead of just isolated features.
- Scenarios describe who the user is, what they want to achieve, and how they interact with the system.
- Scenarios are popular in interactive system design due to their ability to facilitate quick communication about usage possibilities and concerns among stakeholders.

SCENARIO-BASED DESIGN - **AIRLINE ENTERTAINMENT SYSTEM**

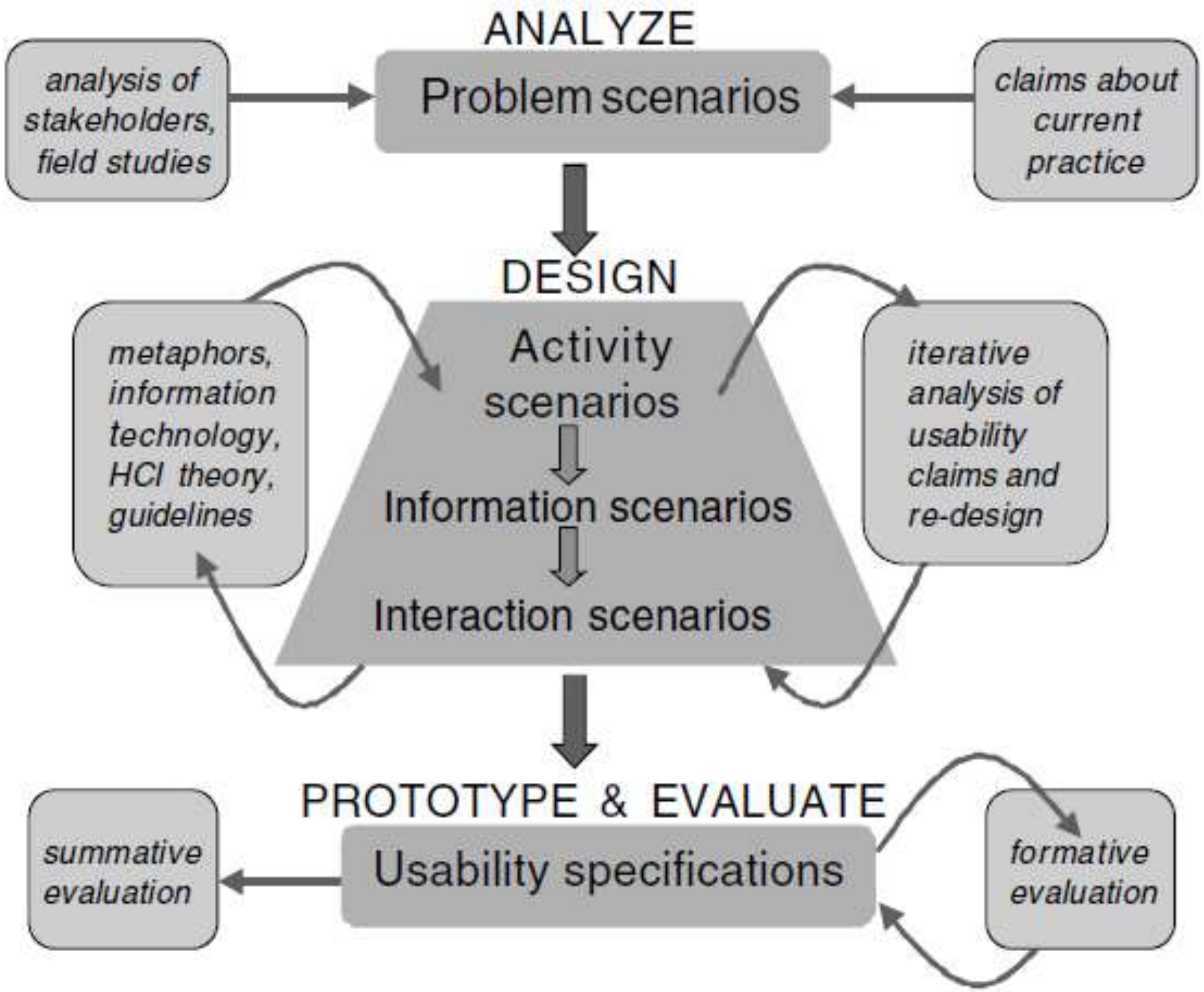
- **Scenario**: Sarah is on a long-haul flight. She wants to watch a movie, track her flight status, and order food via the entertainment screen.

- **Design Use**: Guides the design of a **menu and navigation system** that supports entertainment, travel info, and food ordering.

SCENARIOS IN THE SYSTEM-DEVELOPMENT LIFECYCLE

1. Personas and envisionment
2. Use cases and object-oriented design
3. Analyzing nonfunctional requirements
4. Evaluating systems in use .

SCENARIO-BASED DESIGN (SBD) FRAMEWORK.



POTENTIAL PITFALLS IN SCENARIO-BASED DESIGN

1. Scenario envisioning and analysis can address solution-first design concerns but may introduce bias and narrow views.
2. Designers may focus on incremental changes rather than innovative ideas.
3. Scenario-based design may also lead to documentation burden as designers maintain and evolve large sets of design scenarios.

PARTICIPATORY DESIGN

1. Participatory design is a collaborative design approach that involves end-users in the design process.
2. Its aim is to create products and services that better meet the needs and expectations of users by applying their knowledge and experiences.
3. Participatory design is built on the principles of collaboration, co-creation, and empowerment.
4. Users contribute to the design process : in brainstorming, sketching, prototyping, testing, and decision-making.
5. The core Idea is "Design with users", not just for them.
6. Promotes democracy in design by ensuring the system reflects real user needs.

KEY ASPECTS OF PARTICIPATORY DESIGN

1. **Inclusion:** Include a diverse range of participants who will be affected by or interact with the product, system or problem that needs to be solved. This includes end-users, designers, developers, domain experts, and other stakeholders.
2. **Collaboration:** Involve collaborative activities where participants can contribute their knowledge, insights, and ideas. This collaboration can take various forms, such as workshops, interviews, brainstorming sessions, and co-design exercises.
3. **Empowerment:** Empower users and stakeholders to actively influence design decisions. Their input and feedback are valued and incorporated into the design process, giving them as much ownership and control as designers.

KEY ASPECTS OF PARTICIPATORY DESIGN

1. **Iteration:** Iterate; the design process is iterative, with continuous feedback and refinement. Participants should help evaluate prototypes, provide feedback, and suggest improvements. This iterative approach helps ensure that the final design meets user needs effectively.
2. **Contextual understanding:** Listen to the participants to understand the context in which the final product or system will be used. Learn the cultural, social, and environmental factors to create solutions tailored to their specific context.
3. **User advocacy:** Allow users to advocate for themselves throughout the design process. Address power imbalances and ensure design decisions prioritize users' interests and goals. The goal is to create an equal and safe space for collaboration and co-design.

THE BENEFITS OF PARTICIPATORY DESIGN

1. Improved user satisfaction
2. Increased user engagement
3. Reduced development costs
4. Improved innovation
5. Increased social inclusion
6. Reduced Rework

EXAMPLE

- **IDEO's Design Thinking Process:** IDEO, a global design and innovation consultancy, employs a Participatory Design approach known as Design Thinking. They involve stakeholders and end-users in co-design workshops and prototyping sessions to develop innovative solutions to complex challenges.
- **OpenIDEO:** OpenIDEO is an online platform that facilitates collaborative design challenges, where participants from around the world come together to co-create solutions to social and environmental issues.

HCI IN HEALTHCARE

1. Human-Computer Interaction (HCI) in healthcare focuses on the design, evaluation, and implementation of interactive systems to improve patient outcomes, streamline workflows, and enhance the overall experience for patients and healthcare providers.
2. It encompasses a range of technologies, from electronic health records (EHRs) and wearable devices to telemedicine platforms and decision-support systems.

HCI IN HEALTHCARE

- Smart Clinical Decision Support: Real-time alerts for medication errors, abnormal lab values, or critical situations at bedside.
- Telemedicine Platforms: Live video consultation interfaces optimized for usability and accessibility.
- Wearable Monitoring: Interfaces for real-time patient monitoring (e.g., heart rate, blood oxygen) with actionable alerts for patients/caregivers/physicians.
- Operating Room Dashboards: Touchscreen or hands-free displays for surgeons providing dynamic info (vitals, images) without breaking sterility.

CHALLENGES

1. **Interoperability**: Ensuring systems can communicate and share data effectively.
2. **Privacy and Security**: Protecting sensitive patient information.
3. **Accessibility**: Designing inclusive systems that cater to diverse populations.
4. **Cognitive Load**: Avoiding overwhelming users with too much information.

HCI IN AEROSPACE

- Glass Cockpits: Real-time avionics displays integrating navigation, weather, and system health with intuitive touch/multimodal interaction.
- Mission Control: Live monitoring of spacecraft, drones, and satellites with user-centered visualization and alerting.
- Air Traffic Control (ATC): Large-scale, real-time human-in-the-loop systems for conflict detection and resolution, optimized for cognitive load.
- Fault/Hazard Warning Interfaces: Real-time status and troubleshooting for pilots/astronauts (e.g., NASA, SpaceX Crew Dragon).

HCI IN AEROSPACE

1. **Cockpit Design and Avionics**

- **Goal:** Create intuitive and ergonomic interfaces to ensure pilots can manage complex tasks safely and effectively.
- **Examples:**
  - Glass cockpits with digital displays that adapt dynamically to changing conditions.
  - Heads-Up Displays (HUDs) that project critical flight information onto the windshield.

HCI IN AEROSPACE

2. Automation and Autonomy

- **Focus:** Balancing automated systems with human control to avoid over-reliance or loss of situational awareness.
- **Examples:**
  - Autopilot systems that allow for manual override when necessary.
  - Collaborative control systems in UAVs (Unmanned Aerial Vehicles).

3. Air Traffic Management (ATM)

- **Goal:** Design systems for managing airspace efficiently while reducing controller workload.
- **Examples:**
  - Decision-support tools for conflict detection and resolution in air traffic control.
  - User-friendly radar and communication systems.

HCI IN AEROSPACE

4. Human Factors in Space Exploration

- **Focus:** Supporting astronauts in microgravity environments and during long-duration missions.
- **Examples:**
  - Touchscreen controls designed for gloved hands in spacecraft.
  - AR systems for maintenance tasks aboard the International Space Station (ISS).

5. Flight Training and Simulation

- **Application:** Train pilots and crew in safe, controlled environments using virtual reality (VR) and full-motion simulators.
- **Advances:** Realistic environments that adapt based on trainee responses to simulate emergencies or new systems.

HCI IN AEROSPACE

6.Mission Control Systems

- **Purpose:** Allow ground operators to manage spacecraft and coordinate with crews effectively.
- **Example:** NASA's adaptive interfaces for monitoring space missions in real-time.

7. Passenger Experience

- **Focus:** Designing systems that improve the comfort and safety of passengers.
- **Examples:**
  - In-flight entertainment and connectivity (IFE&C) systems.
  - Cabin safety systems with intuitive emergency instructions.

HCI IN AEROSPACE

8. Augmented Reality (AR) and Virtual Reality (VR)

- **Applications:**
  - AR systems that assist in maintenance by overlaying procedural steps on aircraft components.
  - VR environments for flight rehearsal or space mission planning.

9. Data Visualization and Decision Support

- **Focus:** Presenting complex aerospace data in a digestible format.
- **Examples:**
  - Real-time weather overlays for pilots.
  - Heatmaps for air traffic density to help controllers plan routes.

CHALLENGES IN HCI FOR AEROSPACE

1. **Cognitive Load**: Ensuring users are not overwhelmed with excessive or irrelevant information.
2. **Usability in Extreme Conditions**: Interfaces must perform under stress, high G-forces, or zero gravity.
3. **System Interoperability**: Seamless communication between heterogeneous systems.
4. **Error Prevention**: Designing systems that mitigate human errors, especially in safety-critical operations.
5. **Human-Automation Interaction**: Ensuring trust and clear communication between humans and automated systems.

KEY PRINCIPLES FOR HCI DESIGN FOR KIDS

1. **Simplicity and Clarity**: Interfaces should have minimal clutter, clear navigation, and intuitive interaction paths. **Example**: Large, colorful buttons with recognizable icons.
2. **Engagement Through Play**: Incorporating playful elements, like animations and game-like features, helps maintain attention. **Example**: Interactive storybooks with touch-based elements.
3. **Age-Appropriate Design**: Tailor designs based on developmental stages:
   1. **Preschoolers (3-5 years)**: Simple visuals, voice instructions, and single-task interactions.
   2. **School-aged kids (6-12 years)**: Gamified interfaces and more complex problem-solving tasks.

KEY PRINCIPLES FOR HCI DESIGN FOR KIDS

1. **Accessibility and Inclusivity:** Interfaces must cater to diverse abilities and backgrounds. **Example**: Audio feedback for visually impaired kids or multilingual support.
2. **Physical and Cognitive Development:** Design considering limited dexterity and shorter attention spans. **Example**: Touchscreen gestures like swiping are more natural than precise taps.
3. **Parental Control and Privacy:** Include features that allow guardians to monitor and limit usage. **Example**: Time management tools and content filters.
4. **Safety and Security:** Protect kids from harmful content, data breaches, and online predators. **Example**: COPPA-compliant apps that do not collect personal information.

HCI FOR KIDS

- Educational Games: Interactive, adaptive learning platforms that give kids immediate feedback as they solve puzzles, practice math, or code.
- Augmented Reality (AR) Apps: Real-time object manipulation, collaborative play, or story creation using physical-digital blends (e.g., Osmo).
- Communication Aids: Real-time speech generating and symbol-based apps for children with communication disabilities.
- Safety and Parental Tools: Real-time GPS or activity tracking interfaces with instant alerts and child-centered privacy.

APPLICATIONS OF HCI FOR KIDS

1. **Educational Technology**

- **Examples:** Interactive learning apps, virtual classrooms, and STEM-focused games.
- **Features:** Reward systems to motivate learning and adaptive difficulty levels.

2. Gaming and Entertainment

- **Examples:** Mobile games, augmented reality toys, and virtual reality experiences.
- **Features:** Story-driven content, creative challenges, and multiplayer options.

APPLICATIONS OF HCI FOR KIDS

3. Creative Tools

- **Examples:** Drawing apps, music composition software, and coding platforms.
- **Features:** Drag-and-drop interfaces and visual programming tools.

4. Assistive Technologies

- **Examples:** Communication devices for kids with speech difficulties.
- **Features:** Symbol-based communication interfaces and speech synthesis.

5. Health and Wellness Tools

- **Examples:** Apps promoting physical activity or mental health awareness.
- **Features:** Interactive exercises, trackers, and mindfulness games.

CHALLENGES IN HCI FOR KIDS

1. **Balancing Fun and Functionality**: Ensuring the interface is entertaining without compromising usability or learning outcomes.
2. **Short Attention Spans**: Keeping kids engaged with interactive and visually stimulating elements.
3. **Device and Screen Dependency**: Encouraging healthy usage habits and balancing screen time with offline activities.
4. **Cognitive and Physical Constraints**: Designing for varying levels of literacy, motor skills, and memory capacity.

REVA UNIVERSITY

Thank You