An overview of design concepts and process

COM EM757

Dr. Ayse D. Lokmanoglu

September 16th, 2024

Recap from last week

- Interactivity is important
- We will learn how to communicate with the WWW!
- Glimpse to HTML & CSS

Today

We will talk about design concepts & process

Key Principles of Interaction Design

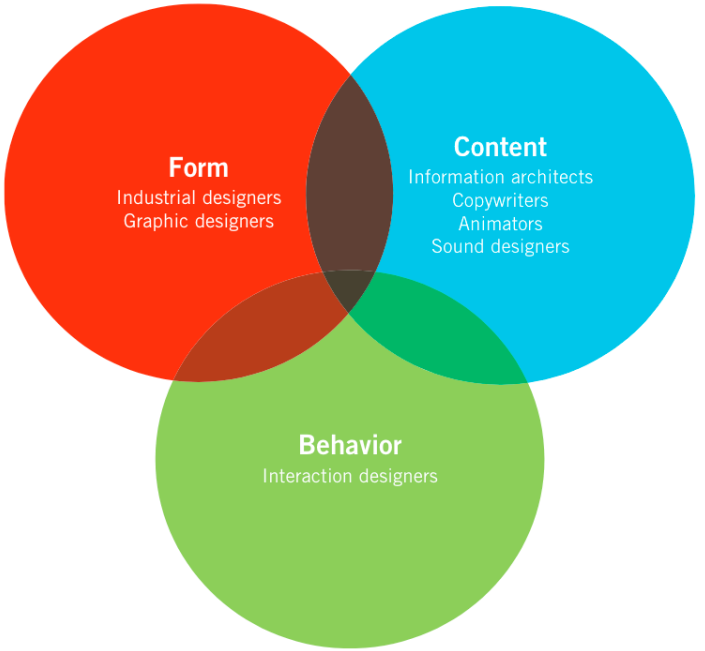


Figure 1: User experience (UX) design has three overlapping concerns: form, behavior, and content. Interaction design focuses on the design of behavior but also is concerned with how that behavior relates to form and content. Similarly, information architecture focuses on the structure of content but also is concerned with behaviors that provide access to content and how the content is presented to the user. Industrial design and graphic design are concerned with the form of products and services but also must ensure that their form supports use, which requires attention to behavior and content.

Cooper et al., 2014, pg. xxiii

Let's talk history

- Desing: *"the conscious and intuitive effort to impose meaningful order"* (Cooper et al., 2014, pg. 3)
- The term "interaction design" - by Bill Moggridge and Bill Verplank in the 1980s.

- While working on the first laptop computer, the GRiD Compass.

- The field remained relatively obscure until the late 1990s when digital product design gained prominence.

- The 1990s web boom drove user-centered design adoption.
- New roles like **interaction designer** and **information architect** emerged.
- "UX" became a key focus as usability gained importance.

Let's go back to complexity

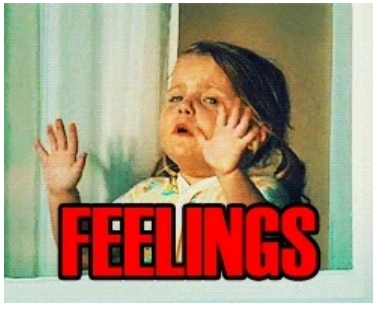
- Interactive products manage thousands of states, unlike mechanical ones.
- Designers must create intuitive systems.
- A structured design process simplifies user interactions.

Humanistic Approach

via GIPHY

Needs, desires, emotions & design

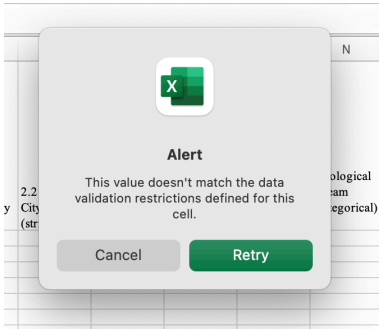
- Interaction design focuses on users' **needs and desires**.
- Designers must understand both logical and emotional user narratives.
- Success depends on meeting functional and emotional needs.



via GIPHY

Consequences of Poor Product Behavior

Digital products are rude



DP need you to think like compters

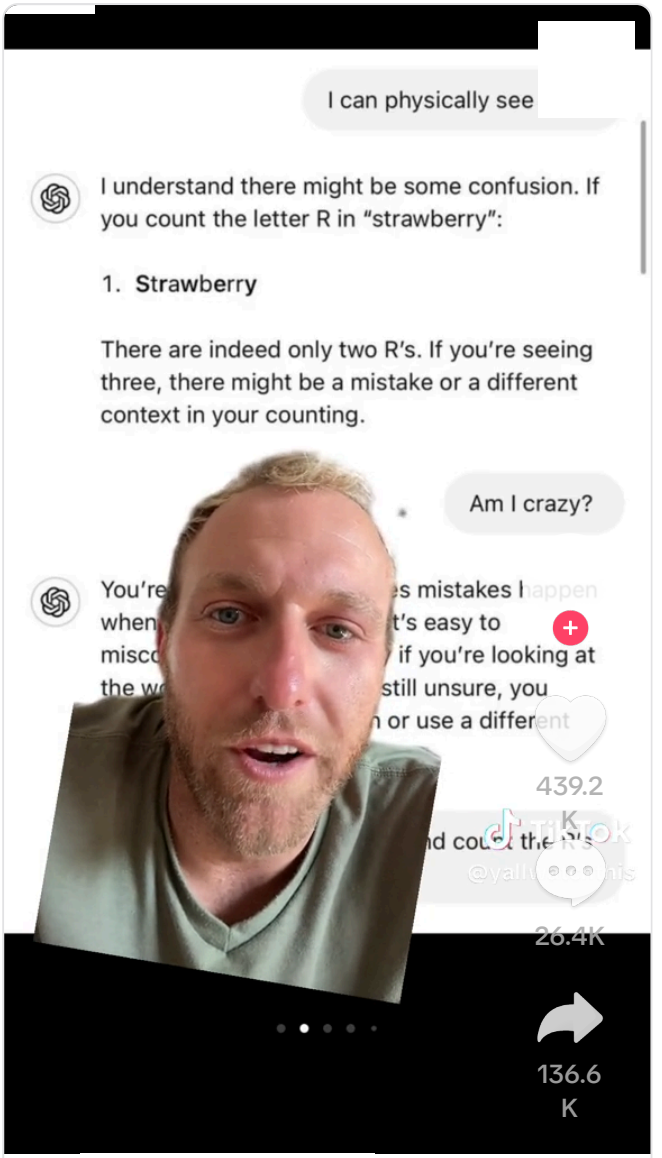
- They assume people are **technology literate** e.g., Save As....

DP have sloppy habbits

e.g., Printing as changing a document?

DP's NEED humans to do the work

e.g., Talking between softwares



@yallwatchthis

Why do digital products fail?

1.Misplaced priorities

Not taking in account users' goals, needs or motivations.

Any examples?

e.g., Nike fuel band

Launched in 2012, discontinued in 2014

- Provided users with abundant data on calories burned and steps taken,
- But its limited features failed to help them understand how those numbers fit into the bigger picture of their overall goals.

What succeeded?

Apple watch? Fitbit?

In pairs let's discuss why they succeeded?

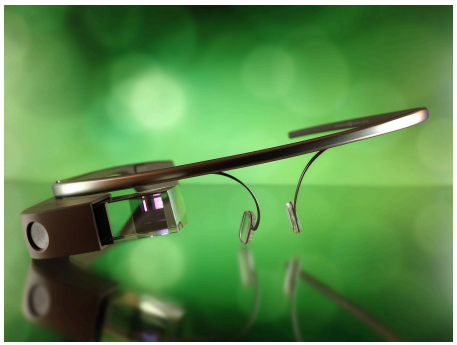
5 minute timer

2.Ignorance about real users

Understand your users

e.g., Google Glass

Launched in 2013, stopped production in 2015.



- Users were uncomfortable with privacy concerns, feeling uneasy about being recorded without consent.
- The design was seen as intrusive and geeky.
- Google failed to consider how Glass would fit into daily life, with most users lacking a reason to wear it regularly.

3.Conflicts of interest

People designing a product should not the same people building it.

e.g., Windows Vista

Launched in 2007, fully discontinued in 2017.

- Windows Vista was developed with a focus on technical implementation over user experience, prioritizing features like enhanced security and graphical upgrades.
- These technical choices made the system slow, resource-heavy, and difficult to navigate, negatively impacting user experience.
- Developers' control over both design and implementation led to a bloated, user-unfriendly product, resulting in widespread frustration and making Vista one of Microsoft's most notable failures.

4.Lack of a design process

The product should follow a reliable process that combines technical feasibility, commercial viability, and user-centered design to meet professional, personal, and emotional needs.

e.g., Google+

Launched in 2011, discontinued in 2019.

- The platform was designed with features like "Circles" for managing friend groups, but it failed to resonate with users due to its confusing interface and lack of clear advantages over established networks.
- Google focused more on integrating the platform across its services rather than understanding what users wanted from a social network, leading to poor adoption.

Summary - why do DP fail

1. Misplaced priorities

1. Ignorance about real users

1. Conflicts of interest

1. Lack of a design process

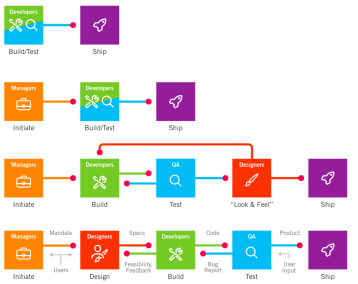


Figure 1-2: The evolution of the software development process. The first diagram depicts the early days of the software industry, when smart developers dreamed up products and then built and tested them. Inevitably, professional managers were brought in to help facilitate the process by translating market opportunities into product requirements. As depicted in the third diagram, the industry matured, and testing became a discipline in its own right. With the popularization of the graphical user interface (GUI), graphic designers were brought in to create icons and other visual elements. The final diagram shows the Goal-Directed approach to software development, where decisions about a product's capabilities, form, and behavior are made before the expensive and challenging construction phase.

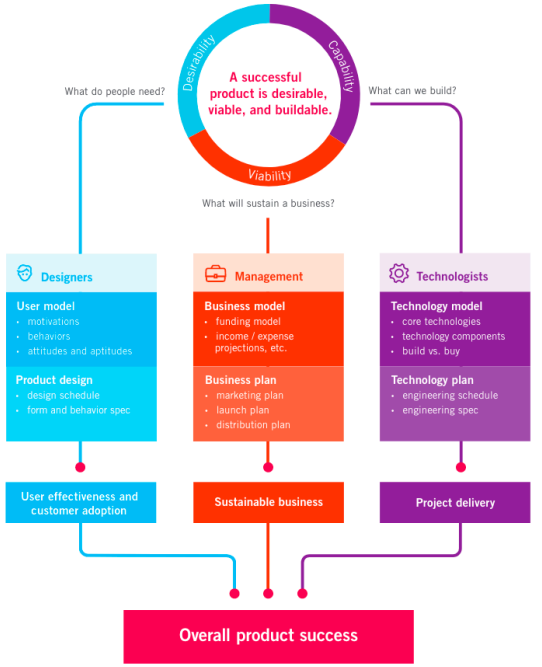
Cooper et al., 2014, pg. 8

Behavior

- Unlike traditional design disciplines, interaction design is centered on the **behavior of digital products**.

- It addresses how products respond to **user inputs**, considering both user needs and system responses.

- Interaction design deals with **designing the user's experience** while **interacting** with a product, emphasizing usability and ease.



You can apply this to companies that have struggled to find the balance:

Apple	Microsoft	Novell
Apple has emphasized desirability but has made many business blunders. Nevertheless, it is sustained by the loyalty created by its attention to user experience.	Microsoft is one of the best run businesses ever, but it has not been able to create highly desirable products. This provides an opening for competition.	Novell emphasized technology and gave little attention to desirability. This made it vulnerable to competition.

Figure 1-3: Building successful digital products. Three major processes need to be followed in tandem to create successful technology products. This book addresses the first and foremost issue: how to create a product people will desire.

Cooper et al., 2014, pg. 12

Complex Behaviors

- Modern digital products - have unpredictable behaviors due to complex internal logic.

- e.g., modern ovens with buttons for multiple functions versus the simpler behavior of older mechanical ovens.

- The shift from physical to digital controls in devices has increased the complexity users must navigate.

Recognizing User Goals

- products designed and built for business goals **FAIL**

Cooper et al., 2014, pg. 13:

Most commercially available software, websites, and digital products often fail to meet user goals by:

- Making users feel frustrated
- Leading to significant mistakes
- Requiring excessive effort to operate
- Lacking an engaging or enjoyable experience

Goals vs tasks & activities

Goals	Tasks and Activities
Desired end state or outcome	Intermediate steps toward achieving a goal
Change slowly over time	Transient and often dependent on available technology
Driven by human motivations	Defined by the tools or technology at hand
Focus on user expectations and aspirations	Focus on the specific actions users take
Provide insight into the meaning of activities	Are steps composed of actions and operations
Help eliminate unnecessary tasks with better technology	May become irrelevant with advancements in technology
Essential for creating designs that satisfy users	Risk limiting designs if focused only on outdated methods

ACTIVITY CENTERED DESIGN

The Activity Centered Design model is an X-Ray into the social and technical workings of an activity. It considers the broader system beyond a single user.

The model is an interconnected lattice with a node for each element. Considering the specifics of these nodes over time is key to developing insight. Read more at dermotholmes.com.

ACTIVITY	**WHAT?** Listen to music while driving.
MOTIVATION	**WHY?** Bordem, mood, social norms.
OUTCOME or GOAL	**DESIRED STATE** Appropriate music is playing.
OBJECT or RECEIVER	**WHAT IS OBSERVED?** The Car.
SUBJECT or CREATOR	**WHO IS DOING THE ACTIVITY?** Driver, Passengers
COMMUNITY	**WHO ELSE?** Radio station, other drivers
DIVISION OF LABOUR	**WHO DOES WHAT?** Passengers may also operate radio.
RULES & RITUALS	**REQUIREMENTS, SOCIAL NORMS** Go to favourite channel first. Polite to ask passengers for input. Minimal distraction to operate.
TOOLS & ARTIFACTS	**THINGS** Interface (shared with GPS) Interface may be bluetooth tablet

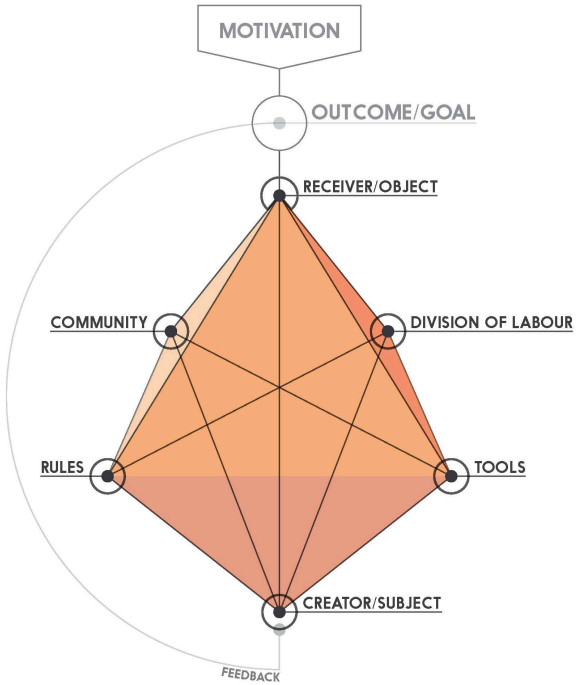


Image from Dermot. (2020, February 20). Activity Centred Design. Dermot Holmes. https://medium.com/dermot-holmes/activity-centred-design-dd28ed1eec59

A good design makes users more effective

Cooper et al., 2014, pg. 17

Who?

Most important users.

What?

Their goals are

Why?

Implementation Models

The implementation model describes how software is built in code, often reflected in the design.

Machines, have mechanisms with moving parts to achieve their purpose, e.g., projector.

- Developers may design interfaces based on this model, creating buttons, fields, and pages for each function.

- This approach, while **logical for developers**, often confuses users and hinders their ability to achieve their goals.

Mental Models

Mental models allow users to interact with technology without needing to understand its inner workings, but discrepancies with implementation models are common, especially in software.

People create cognitive shortcuts, or mental models, to explain how things work without understanding their inner mechanics.

- e.g,, many imagine electricity flows like water through cords, even though this doesn't reflect the actual process.
- In the digital world, mental models often differ greatly from implementation models, as seen with cell phones, which function as radio transceivers.
- The complexity of software makes it difficult for users to connect their actions to the application's internal processes.

e.g. Search Engines

Implemented Model: Search Engine Algorithm

- In a search engine like Google, the implemented model is the complex system of algorithms and indexing processes that retrieve and rank web pages. This includes crawling websites, storing data in massive databases, and applying algorithms like PageRank to sort search results based on relevance, backlinks, user engagement, and hundreds of other factors. This technical system is highly complex and operates behind the scenes.

Mental Model: User's View of a Search Engine

- From the user's perspective, the mental model is much simpler. They believe that typing a query into a search bar retrieves the most relevant websites almost instantaneously. The user is unaware of the crawling, indexing, and ranking algorithms; they assume the search engine just *finds* the best information quickly, almost as if it’s pulling directly from a database of perfect answers.

The implemented model (complex algorithmic processes) is **hidden from the user**, while the mental model (simple search-and-find process) makes **interacting with the system intuitive and easy**.

Represented Models

The "represented model" is how designers choose to present the software's operation to users, which can differ from the actual processing structure.

Figure 1-4: A comparison of the implementation model, mental model, and represented model. The way engineers must build software is often a given, dictated by various technical and business constraints. The model for how the software actually works is called the *implementation model*. The way users perceive the jobs they need to do and how the application helps them do so is their *mental model* of interaction with the software. It is based on their own ideas of how they do their jobs and how computers might work. The way designers choose to represent the working of the application to the user is called the *represented model*. Unlike the other two models, it is an aspect of software over which designers have great control. One of the designer's most important goals should be to make the represented model match a user's mental model as closely as possible. Therefore, it is critical that designers understand in detail how their target users think about the work they do with the software.

"User interfaces should be based on user mental models rather than implementation models."

Cooper et al., 2014, pg. 19

Behavior, models, needs, motivations and goals

- Software has a "behavioral face" shown to users, often different from how the program truly functions.
- Aligning the represented model with the user's simpler mental model makes the software easier to use and understand.
- A represented model closely following the implementation model can confuse users, as their mental model is typically much simpler than the underlying technical structure.
- Effective software design simplifies complex data and operations into a form that aligns with users’ mental models.

In pairs let's think about another example than Adobe that is a good example of a represented model.

e.g., Apple's iOS calculator

- Users see buttons for numbers and basic operations, which align with their mental model of how a calculator should work.
- Behind the scenes, the calculator is performing complex computations, but the user only interacts with a clean, intuitive interface that hides unnecessary technical details.
- By aligning the represented model with the user’s mental model, Apple makes the calculator easy to use, even for people who are not familiar with advanced mathematical operations or computing processes.

"Goal-directed interactions reflect user mental models." (Cooper et al., 2014, pg. 20)

Why?

- Many tech companies lack a formal design process.
- Even established processes struggle with research-to-design gaps.
- Market research helps sales but doesn't reveal user behavior with complex products.
- Traditional methods fail to translate research into design solutions.
- Goal-Directed methods address the gap between user research and product design.

Goal-Directed Design (GDD) method

- Goal-Directed Design focuses on users' goals, expectations, and skills.
- It aims to create **powerful, effective, and enjoyable** solutions.
- GDD aligns product design with both **user** and **business** goals.

Concept	Key Points	Goal-Directed Design Approach
Design as Product Definition	Design should define both behavior and appearance, aligning with user goals, business needs, and technology constraints.	Goes beyond a visual facelift to provide true product definition, focusing on user-centered requirements.
Designers as Researchers	Designers should engage in research to build empathy and ensure their designs meet real user needs.	Involving designers in research closes the gap between user behavior and design, preventing isolation from users.
Bridging the Gap	Traditional methods fail to translate research into actionable design, focusing too much on tasks rather than user goals.	Goal-Directed Design creates systematic processes that translate user research into design specifications and frameworks.

6 Phases

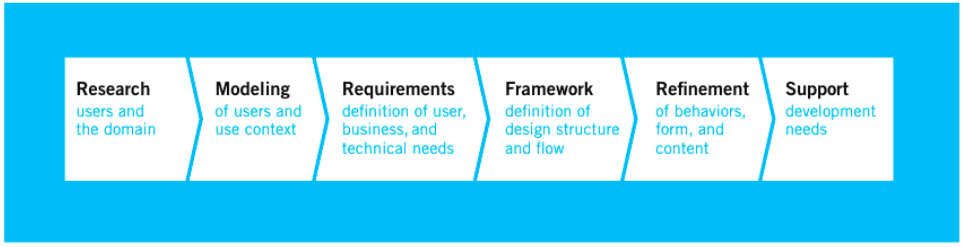


Figure 1-7: The Goal-Directed Design process

1.Research,

2.Modeling,

3.Requirements Definition,

4.Framework Definition,

5.Refinement, and

6.Support.

1.Research

- Research phase uses ethnographic studies, competitive audits, market research, and interviews with stakeholders, developers, and experts.

- Field observations and interviews identify behavior patterns, which are linked to goals and motivations, and inform persona creation in the Modeling phase.

- Market research filters personas, aligning them with business models, while stakeholder interviews and product audits reveal business goals and technical constraints.

2.Modeling

- Synthesizes behavior and workflow patterns into domain and user models (personas).
- **Personas**: Detailed user archetypes representing behaviors, attitudes, and goals, used for scenario-based design.
- Personas help define and iterate on design concepts, ensuring coherence and alignment with user needs.
- Designers prioritize personas by comparing their goals, ensuring no gaps or duplications.
- Personas guide design targets and influence the final product form and behavior.

3.Requirements

- Connects user models to design frameworks using scenario-based methods focused on persona goals.
- Personas drive the identification of key tasks and inform interface design to minimize effort and maximize return.
- The design process involves analyzing persona data, functional needs, and interactions via iterative context scenarios.
- Considers personas' skills, physical capabilities, and environment, along with business goals and technical constraints.
- Output: A requirements definition balancing user, business, and technical needs.

4.Framework

- Defines the overall product concept, including behavior, visual design, and physical form (if applicable).
- Interaction Framework: Synthesized using interaction principles and design patterns.
- Interaction design principles guide appropriate system behavior across contexts.
- Interaction patterns provide general solutions to recurring problems, based on proven design knowledge.
- Translates functional needs into design elements, organized into sketches and behavior descriptions.
- Iterative scenarios refine details in the Refinement phase, balancing top-down (pattern) and bottom-up (principle) design.
- Collaboration with industrial designers ensures interaction concepts work with physical prototypes.
- Collaboration with service designers drafts service maps and blueprints for both user and provider experiences.
- Visual designers develop visual frameworks (visual language strategy) based on brand attributes and interface structure.

5.Refinement

- Interaction Designers:
  - Ensure task coherence through key path and validation scenarios, using detailed storyboards.

- Visual Designers:
  - Define a system of type styles, icons, and visual elements, ensuring clear **affordances and hierarchy**.

- Industrial Designers:
  - Finalize materials and collaborate with engineers on assembly and technical issues.

- The outcome is a form and behavior specification or blueprint, delivered in paper or interactive media as needed.

6.Support

- Interaction designers must remain available to assist developers during the construction process.
- Development challenges or timeline pressures may require scaled-down design solutions.
- Without design team input, developers may compromise design integrity when making trade-offs.
- Ongoing collaboration ensures the design's integrity is preserved throughout development.

via GIPHY

Goal-Directed Design Process Overview (Cooper et al., 2014, pg. 25)

Phase	Activity	Concerns	Stakeholder Collaboration	Deliverable
Research	Scope	Objectives, timelines, financial constraints, process	Meetings (Capabilities & Scoping)	Statement of Work
	Audit	Branding strategy, market research, competitors	Interviews (Stakeholders & Users)	Market Research Review
	Stakeholder Interviews	Product vision, risks, constraints, users	Interviews (Stakeholders & Users)	Stakeholder Insights
	User Interviews & Observations	Understand user needs and behavior	Check-in (Preliminary Research)	Research Findings
Modeling	Personas	User behavior patterns, attitudes, goals	Check-in (Personas)	Persona Development
	Other Models	Workflows among users and environments	Check-in (Models)	User and Environment Models
Requirements Definition	Context Scenarios	Ideal user experiences, aligning with user goals	Check-in (Scenarios)	Scenarios and Requirements
	Requirements	Necessary product capabilities	Presentation (User & Domain)	User and Domain Analysis
Design Framework	Elements	Manifestation of functionality and information	Check-in (Framework)	Design Framework
	Framework	Object relationships, flow, navigation	Presentation (Design Vision)	Design Vision
	Key Path & Validation Scenarios	Interaction sequence and user behavior	Check-in (Validation Scenarios)	Design Specification
Design Refinement	Detailed Design	Interface details, visualization, experience, behavior	Check-in (Design Refinement)	Form and Behavior Specification
Support	Design Modification	Accommodating new constraints	Collaborative Design	Revision of Specifications

5 minutes in paris let's discuss GDD and try to come up with examples.

Let's return to Interaction Design vs. UX Design

Interaction Design

- Focuses on behavior
- Emphasizes behavior
- Crucial role in creating the experience

UX Design

- Covers the whole experience
- Includes form, function, content
- Interrelated with interaction design