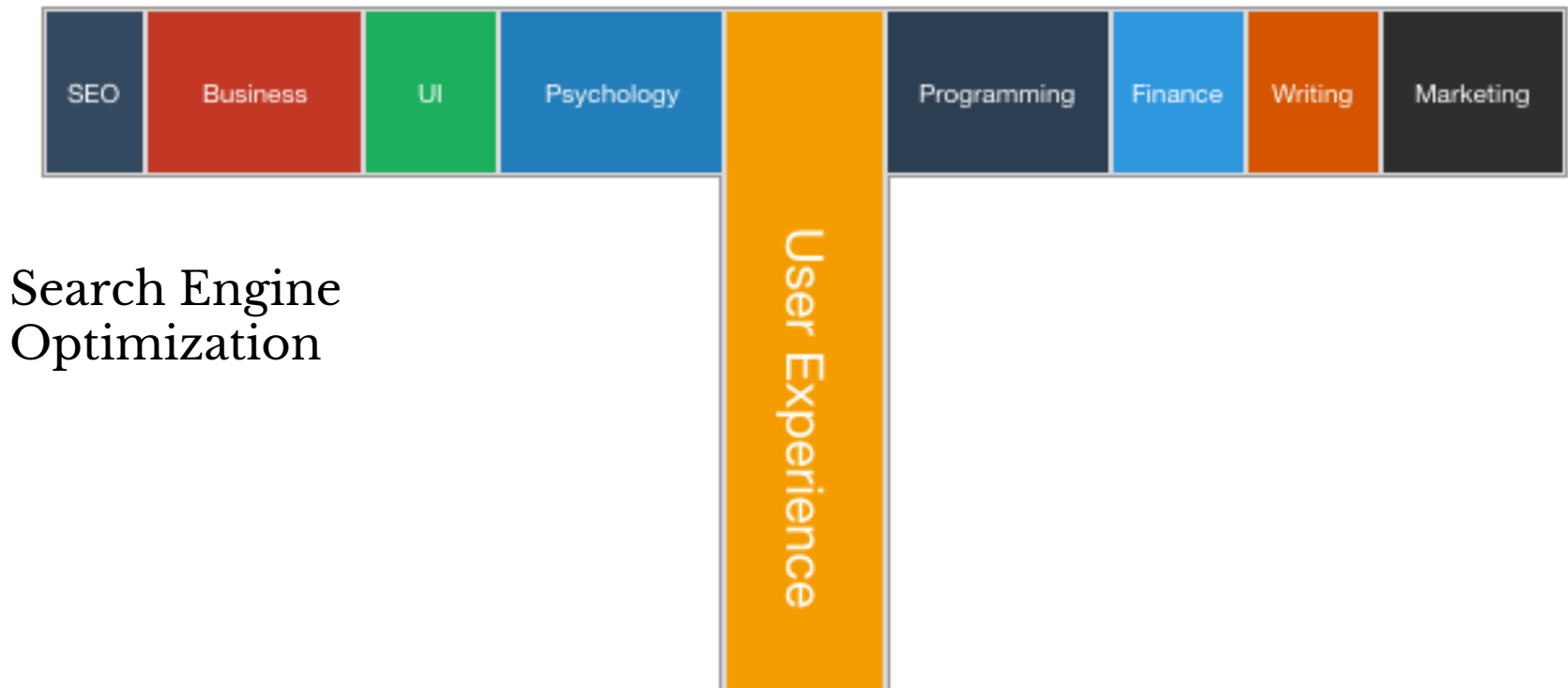


User Experience Design

Who are the designers and what are their roles?

- Profiles are T-shaped- they have a T-shaped thought process and the expertise is T-shaped.
- They have a horizontal breadth of



Who are the designers and what are their roles?

- Must understand the following:
 - User's Need
 - Web Design
 - Market Requirements(Management People)

UX Designers ,UI Designers, Visual Communication Designers

- UI – design the Interface
- UX- They depict the **experience of the user, when they use the interface**
- UX- how people behave when they are given that product/ interface to use.
- More of a research profile
- They will survey/research about the user's experience with the current system/product
- Based on this the User interface has to be designed.(structure/ layout of the design)

UX Designers ,UI Designers, Visual Communication Designers

- Visual Communications Designers-
 - Mostly design the Graphical Content
 - create the detailing with the color, texture and how the UI transition will happen
 - Typography, the shape and size of the tabs and the buttons and all this color palette and other detailing
 - May design Animations

User Interfaces Are Hard to Design

- **You are not the user**

- Most software engineering is about communicating with other programmers
- UI is about communicating with users

- **The user is always right**

- Consistent problems are the system's fault, If users consistently make mistakes with some part of your interface, take it as a sign that your interface is wrong, not that the users are dumb.

- **But the user is not always right**

- Users aren't designers. E.g. (lighter hand sets)

UX Designers ,UI Designers, Visual Communication Designers- The Design Team

- A close communication between the user experience designer people and the management.
- So they sometimes do the similar task- understanding the user requirement, management people
- They will be in sync with the idea that the product should cater to this kind of people.

● Design team along with the others

Design and Research Methods

- Historical
- Qualitative
- Co relational
- Experimental
- Simulation
- Logical/Argumentation

Linda Groat and David Wang,
Architectural

Design and Research Methods

- **Historical research**- we dig down what happened in the past/history.
 - The systematic collection and evaluation of data to describe, explain and understand action or events that occurred sometime in the past.
- **Qualitative Research**-analyze user behavior, what they think, how they behave.
 - In qualitative consist of one-on-one interview, case study, Focus group and record keeping.

Design and Research Methods

- **Co relational Research-** we compare 2 cases, case1 and case 2 and find which is better
 - We give 2 options to the users for the same problem and try to find which one is better

Design and Research Methods

- **Experimental Research**- Some scenario is created in augmented reality which is not there. Experimental it is like virtual setup when we conduct a experiment within a typical virtual setup and we collect the data.
- E.g. how people might behave in a disastrous situation for example evacuation of a flight.
- **Simulation research**- generally does not happen in UI UX design because it might be totally devoid of human behavior.
- So simulation is totally data centric, eye tracking data and this can be as a simulation but there also human

Usability Engineering Is a Process

- Design
- Implement
- Evaluate

1.Design

● Task analysis

● “Know the user”

- Who are they?
- What do they already know?
- What is their environment like?
- What are their goals?
- What information do they need, what steps are involved in achieving those goals?

● Design guidelines

- guidelines are heuristics, not hard-and-fast rules
- Avoid mistakes
- May be vague or contradictory
- Design guidelines help us discuss design alternatives sensibly, but they don't give all

2.Implement

- Prototyping(we can't predict usability in advance, we build prototypes)
- Prototyping is used to cut down on the complexity of implementation by eliminating parts of the full system.
 - Cheap, throw-away implementations
 - Low-accuracy: paper
 - Medium-accuracy: HTML, Visual Basic
- GUI implementation techniques
 - Toolkits

3. Evaluate

- Evaluation puts prototypes to the test
- Expert evaluation
 - Heuristics and walkthroughs
- Predictive evaluation
 - Testing against an engineering model (simulated user)
- Empirical(Experimental) evaluation
 - Watching users do it

Iterative Design

● We don't go around the design-implement-evaluate loop just once.

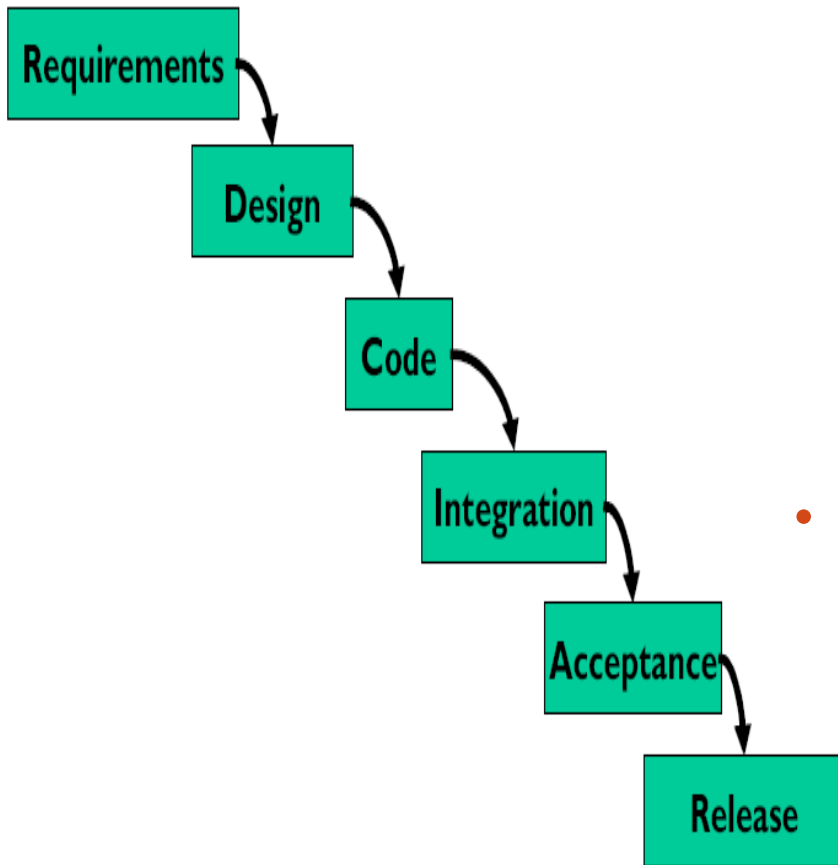
● Using the results of evaluation, we need to redesign the interface, build new prototypes, and do more evaluation.

● You won't get it right the first time.

● Iterative design is the current best-practice process for developing user

- Design
- Implement
- Evaluate

Traditional Software Engineering Process: Waterfall Model



- Each stage results in a concrete product
 - a requirements document,
 - a design,
 - a set of coded modules – that feeds into the next stage.
- Each stage also includes its own validation: the design is validated against the requirements, the code is validated (unit-tested) against the design
- Validation is not always

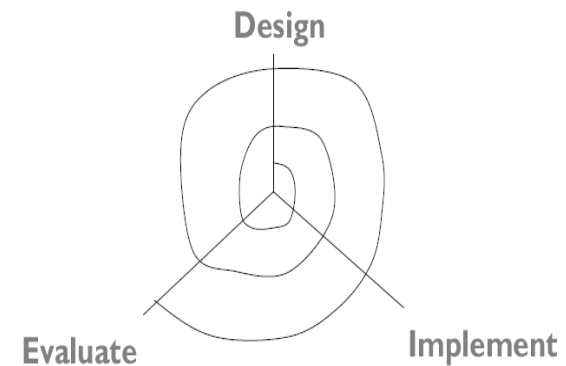
Disadvantage of Waterfall model

- The danger arises when a mistake in an early stage – such as a missing requirement – isn't discovered until a very late stage – like acceptance testing.
- Mistakes like this can force costly rework of the intervening stages. (That box labeled “Code” may look small, but you know from experience that it isn't!)

Waterfall Model Is Bad for UI Design

- User interface design is risky
 - So we're likely to get it wrong
- Users are not involved in validation until acceptance testing
 - users appear in the process in only two places: requirements analysis and acceptance testing
 - So we won't find out until the end
- UI flaws often cause changes in requirements and design
 - So we have to throw away carefully-written and tested code
 - Slapping on patches doesn't fix serious

Spiral Model



- Early iterations as cheap as possible.
- The radial dimension of the spiral model corresponds to the cost of the iteration step – or, equivalently, its accuracy.
- E.g. an early implementation might be a paper sketch or mockup.
- It's low-accuracy, only rough idea of what it would look and behave like as interactive software.
- But it's incredibly cheap to make, and we can evaluate it by showing it to users and

Iterative Design of User Interfaces

- Early iterations use cheap prototypes
 - Risk is greatest in the early iterations, when we know the least.
 - Early prototypes are made to be thrown away
 - Parallel design is feasible: build & test multiple prototypes to explore design alternatives without much expense.
- Later iterations use richer implementations, after UI risk has been mitigated.
 - After we have evaluated and redesigned several times, we have (hopefully) learned enough to avoid making a major UI design error

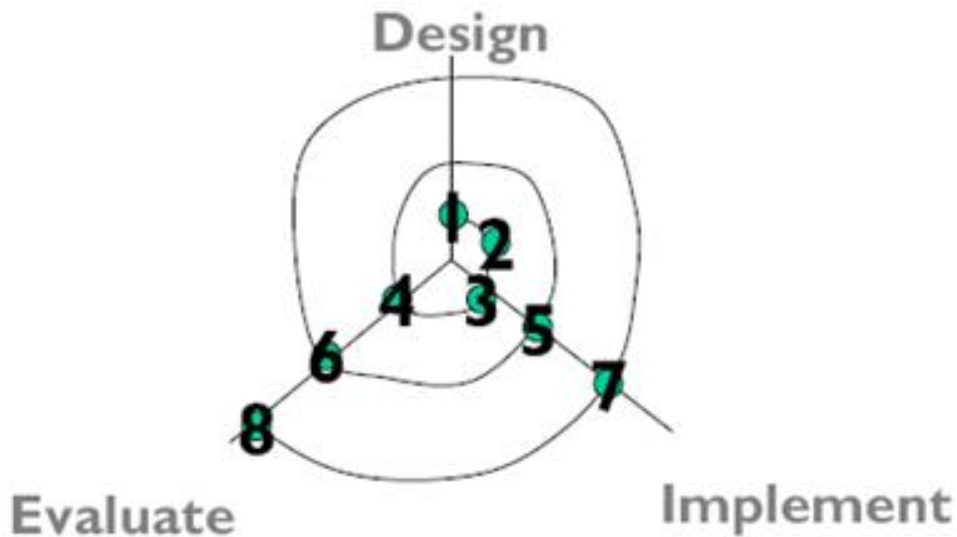
Iterative Design of User Interfaces

- More iterations generally means better UI
 - We keep the parts of the design that work, and redesign the parts that don't.
- Only mature iterations are seen by the world

User-Centered Design

- Iterative design is a crucial part of **user-centered design**.
- The practice of creating engaging, efficient user experiences is called **user-centered design**.
- Take the user into account every step of the way as you develop your product.
- **User-centered Design**- widely accepted among UI practitioners.

User Centered Design



1. Task analysis
2. Design sketches
3. Paper prototype
4. In-class user testing
5. Computer prototype
6. Heuristic evaluation

User & Task Analysis

- The process of collecting information for the first design is called **user and task analysis**.
- First step of user-centered design
- **User analysis:** who is the user?
- **Task analysis:** what does the user need to do?

Know The User

- Identify characteristics of target user population
 - Age, gender, ethnicity
 - Education
 - Physical abilities
 - General computer experience
 - Skills (typing? reading?)
 - Domain experience
 - Application experience
 - Work environment and other social context
 - Relationships and communication patterns -
Can users ask each other for help, or are they isolated? How do students relate differently to lab assistants, teaching assistants, and professors?

Multiple Classes of Users

- Since you're not the user, you need to find out who the user actually is.
- User analysis is often Skipped, which leads to falling into the trap of assuming every user is like you.
- Many applications have several kinds of users
- Example: Hospital Management System
 - Registration Clerks
 - Patients
 - Accounts Clerks
 - System admins
 - Nurses
 - Doctors
- Many applications have to worry about

How To Do User Analysis

● Techniques

- Questionnaires (straightforward characteristics can be obtained)
- Interviews (details about the context and environment)
- Observation (details about the context and environment)

● Obstacles

- Developers and users may be systematically isolated from each other
 - Tech support shields developers from users
 - Marketing shields users from developers
 - Some users are expensive to talk to
 - Doctors, executives, union members

- little money spent collecting information initially

Benefits of User Analysis

- **Better products** — Processes that involve end users as well as understand business objectives will always result in products that work better for their intended purpose.
- **Cheaper to fix problems** — User analysis helps you match up your product against reality to make changes while it's still mostly just on paper.
- A wireframe or prototype is magnitudes cheaper than a technical fix to a live product.
- **Ease of use is a common requirement** — Customers often use the terms “usability” and “user experience” when describing qualities they seek in products.
- Therefore, user analysis drives your product to have better selling points.

Example: Self-Service Grocery Checkout

- Who are the users?
 - Grocery shoppers
 - Wide range of ages (10-80) and physical abilities (height, mobility, strength)
 - No computer experience
 - No training: walk up and use
 - Knowledge of food, but not about supermarket inventory techniques
 - Supermarket shoppers often ask each other for help finding things
- Major user classes
 - Family shopping is often done by women, often accompanied by small children
 - Store clerks who need to help shoppers

Task Analysis

- Each task is a goal (*what, not how*)
- Often helps to start with overall goal of the system and then decompose it hierarchically into tasks
- Overall goal:
 - shoppers pay for their own groceries
- Tasks:
 - Enter groceries into register
 - Bag groceries
 - Pay

Essential Parts Of Task

Analysis

- Every task in a task analysis should have at least these parts

1. Goals
2. Pre Conditions (e.g email.)
3. Subtasks

- What needs to be done?
 - Goal
- What must be done first to make it possible?
 - Preconditions- a. Tasks on which this task depends, b. Information that must be known to the user
 - users don't always satisfy them before attempting a task, resulting in errors.

- What steps are involved in doing the task?

- Subtasks

Example: Self-service Grocery Checkout

- Goal
 - Enter groceries into register
- Preconditions
 - All the groceries you want are in your cart
- Subtasks
 - Enter prepackaged item
 - Enter loose products

Other Questions to Ask About a Task

- Where is the task performed?
 - Front of supermarket, standing up
- How often is the task performed?
 - At most a few times a week
- What are its time or resource constraints?
 - A minute or two
- How is the task learned?
 - By trying it
 - By watching others
 - By being shown how by store personnel
- What can go wrong? (Exceptions, errors, emergencies)
 - Barcode is missing or smudged
 - Shopper wants to buy alcohol or cigarettes
- Who else is involved in the task?

Dangers of Task Analysis

- Duplicating a bad existing procedure in software
 - E.g. “Find page N”
- Failing to capture good aspects of existing procedure - interviewing and observing real users is important
 - E.g. “Important notes on paper”

Hints for Better User & Task Analysis

- Questions to ask
 - Why do you do this? (goal)
 - How do you do it? (subtasks)
- Look for weaknesses in current situation
 - Goal failures, wasted time, user irritation
- Contextual inquiry
- Participatory design

Contextual inquiry

- Contextual inquiry is a technique that combines interviewing and observation, in the user's actual work environment, discussing actual work products.
- – User shows how and talks about it
- – Interviewer watches and asks questions

Participatory Design

- Include representative users directly in the design team
 - participating in the task analysis, proposing design ideas, helping with evaluation.
- This is particularly vital when the target users have much deeper domain knowledge than the design team.
- E.g.
- It would be unwise to build an interface for stock trading without an expert in stock trading on the team.

Other Questions to Ask about a Task

- Where is the task performed?
 - At a kiosk, standing up
- What is the environment like? Noisy, dirty ?
 - Outside
- How often the task is performed?
 - Couples of times a day
- How is task learned ?
 - By trying
 - By watching others
 - Classroom Training?
- What can go wrong?
 - Enter wrong country code
 - Enter wrong user name
 - Get distracted while recording/ listening message
- Who else is involved in the task?

THANK
YOU