Human Computer Interaction
CE 382

Course Instructor: Vincent M. Nofong, Ph.D.

July 9, 2024

Introduction

Outline

- Who I am
- Course Information and Outline of CE 382
- Expected Learning Outcomes
- Rules
- Chapter Two: Establishing Requirements

Introduction

About me

- Name: **Vincent M. Nofong, PhD**
- Email: **vnofong@umat.edu.gh**
- Personal Website: `https://vincentnofong.com/`
- Uni website: `https://www.umat.edu.gh/staffinfo/staffDetailed.php?contactID=385`
- Office hours (Working days): **09:00 am - 16:00 pm GMT**
- Research interest: **data mining, trend prediction, classification, bioinformatics, artificial intelligence, machine learning**

Introduction

Course Information (CE 382)

- Credit hours: **3**
- Attendance: **10%**
- Continuous Assessment: **30%**
  - Quizzes - two or three
  - Group assignment - one (application development)
  - Group presentations
- End of Semester: **60%**

Introduction

Course Outline (CE 382)

1. Interaction Design
2. Establishing Requirements
3. Prototyping
4. Data Gathering and Analysis
5. Cognitive Aspects of Design
6. Social and Emotional Interactions
7. User Interfaces
8. Evaluations

Introduction

Expected Learning Outcomes (CE 382)

Students should understand and be able to:

1. Explain the characteristics of good and bad interaction design and use them to evaluate HCIs
2. Explain the characteristics of users that influence HCI and use them to inform user interface development
3. Explain, analyze and develop interaction evaluations
4. Explain and develop requirements for interaction design
5. Construct interactions using evaluation-based iterative process for directing the design of user interfaces.

Introduction

Reference Materials

1. Preece, J., Rogers, Y. and Sharp, H. (2023), Interaction Design: Beyond Human-Computer Interaction, John Wiley & Sons Ltd, Hoboken, U.S.A., 6th Edition, 716 pp. - slides are based on this reference
2. Lazar, J., Feng, J. H. and Hochheiser, H. (2017), Research Methods in Human-Computer Interaction, Morgan Kaufmann, Burlington, U.S.A., 2nd Edition, 560 pp.
3. Shneiderman B., Plaisant C., Cohen M. and Jacobs, S. (2016), Designing the User Interface, Pearson Publishers, 6th Edition, 616 pp.

Introduction

Rules

1. Feel free to ask questions in class, unless they are too “personal”.
2. Students should not be late for lectures or practicals.
3. Students should attend all lectures and practicals.
4. **In case you are unable to attend lectures or will be late, send me an email - at least 30 minutes before lectures.**
5. Students should do and submit all assignments before the given deadline.
6. Unless otherwise permitted, students should not use their mobile phones in class - note usage of Laptops/Desktops is permitted.

HCI CE 382
Chapter Three: Prototyping

Course Instructor: Vincent M. Nofong, Ph.D.

July 9, 2024

Prototyping

What is a Prototype?

- A prototype is a representation of a design that enables stakeholders to interact with it.
- In various design fields, prototypes can take the form of small-scale models.
- Examples of prototypes include miniature cars, miniature buildings, or miniature towns.
- Prototypes serve as tangible representations that help visualize and evaluate the design concept.
- They allow stakeholders to provide feedback, test functionality, and make informed decisions before final implementation.

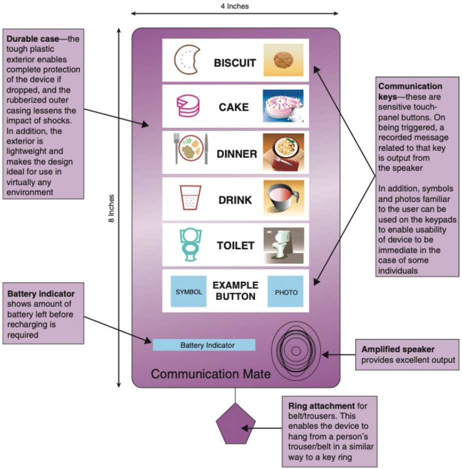
Prototyping

What is a Prototype in Interaction Design?

- In interaction design, a prototype serves various purposes and can take different forms.
  - Examples include series of screen sketches, storyboards, PowerPoint slide shows, video simulations, physical prototypes, loosely connected electronic elements, animations of product use, and software with limited functionality.
- These prototypes help visualize and demonstrate the intended user experience and functionality of the design.
- They allow for user feedback, usability testing, and iterative design improvements before final implementation.
- Choosing the appropriate prototype format depends on the specific project requirements, goals, and resources available.

Prototyping

Example Interaction Design Prototype: A paper-based prototype of a handheld device to support an autistic child



Durable case—the tough plastic exterior enables complete protection of the device if dropped, and the rubberized outer casing lessens the impact of shocks. In addition, the exterior is lightweight and makes the design ideal for use in virtually any environment

Battery indicator shows amount of battery left before recharging is required

Communication keys—these are sensitive touch-panel buttons. On being triggered, a recorded message related to that key is output from the speaker

In addition, symbols and photos familiar to the user can be used on the keypads to enable usability of device to be immediate in the case of some individuals

Amplified speaker provides excellent output

Ring attachment for belt/trousers. This enables the device to hang from a person's trouser/belt in a similar way to a key ring

Prototyping

Why Prototype?

- It facilitates evaluation and feedback in interaction design.
- Prototypes provide a tangible and interactive representation of the design, making it easier for stakeholders to comprehend and engage with.
- Prototypes allow ideas to be tested and explored, helping designers make informed decisions.
- Reflection is encouraged through the prototyping process, promoting critical assessment and refinement of designs.
- Prototypes help answer questions and support designers in choosing between alternative design solutions.

Prototyping

Low-Fidelity Prototyping

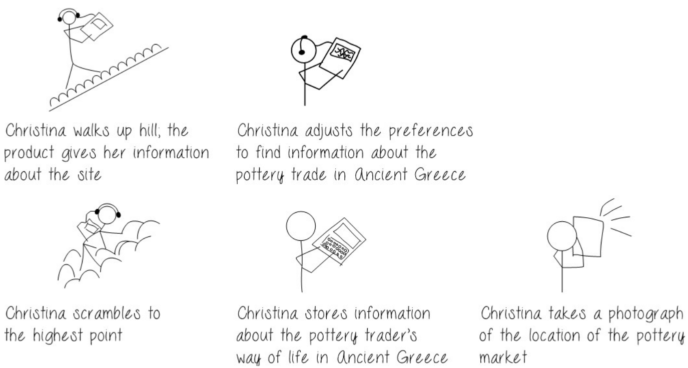
- Low-fidelity prototyping involves using a medium that is different from the final product, such as paper or cardboard.
- It is a simple, quick, and cost-effective method that allows for easy modifications.
- Examples of low-fidelity prototyping include:
  - sketches of screens and task sequences,
  - index cards or sticky notes,
  - storyboards, and,
  - the Wizard-of-Oz technique

Prototyping

Low-Fidelity Prototyping: Storyboards

- Storyboards are a series of sketches that depict the progression of a user through a task using the product.
- They are frequently used in conjunction with scenarios to provide more detail and enable role-playing.
- Storyboards offer a visual representation of the user experience and help in understanding the flow of interaction with the product.

Low-Fidelity Prototyping: Storyboards -An Example



Christina walks up hill; the product gives her information about the site

Christina adjusts the preferences to find information about the pottery trade in Ancient Greece

Christina scrambles to the highest point

Christina stores information about the pottery trader's way of life in Ancient Greece

Christina takes a photograph of the location of the pottery market

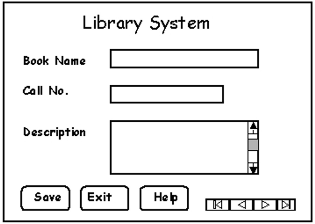
Prototyping

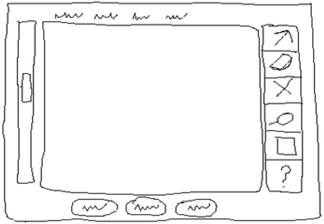
Low-Fidelity Prototyping: Sketching

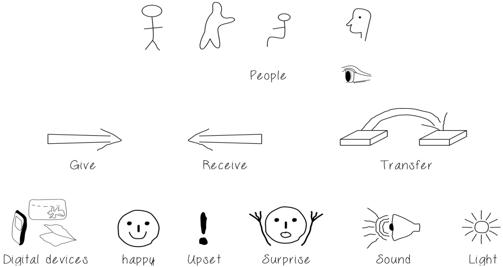
- Sketching is a common technique used in low-fidelity prototyping.
- It allows for quick and easy visualization of design ideas and concepts.
- Drawing ability should not be a limitation, as simple symbols and shapes can effectively communicate design concepts.
- Sketching helps to explore and iterate on design ideas before investing significant time and resources in high-fidelity prototypes.

Prototyping

Low-Fidelity Prototyping: Sketching - Examples







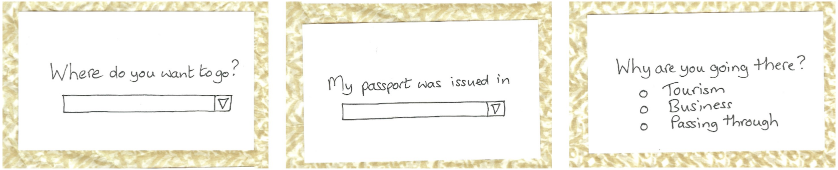
Prototyping

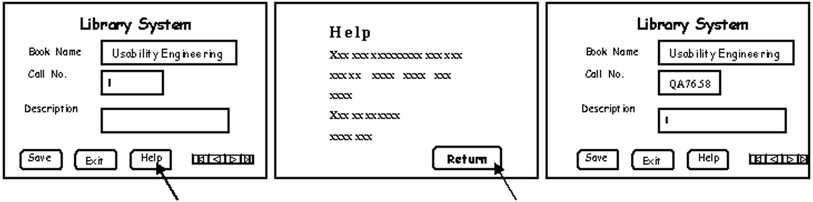
Low-Fidelity Prototyping: Index Cards

- Index cards (3 x 5 inches) can be used for prototyping.
- Each card represents a specific element of interaction in the design.
- During evaluation, the cards can be stepped through to simulate the flow and interaction of the system.
- Index cards provide a tangible and flexible way to prototype and iterate on design ideas.

Prototyping

Low-Fidelity Prototyping: Index Cards - Examples





Don't know how to input → **Help screen for *Call No.* Field** → **Finish input *Call No.***

Click Help → **Click *Return* after reading** → **Ready to input the next field**

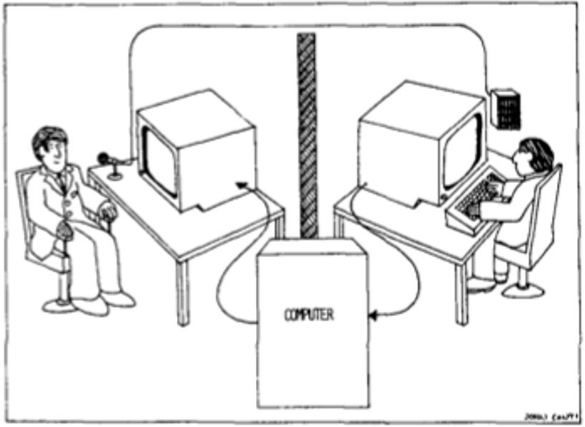
Prototyping

Low-Fidelity Prototyping: Wizard-of-Oz

- Wizard-of-Oz prototyping involves simulating an interactive system where a human responds to the participant's input instead of using an actual functioning system.
- Participants believe they are interacting with a computer, unaware that a human is behind the scenes.
- This technique is commonly used in the early stages of design to gain insights into people's expectations and behaviors.
- Wizard-of-Oz prototyping allows designers to observe user interactions and collect valuable feedback before investing in the development of a fully functional system.

Prototyping

Low-Fidelity Prototyping: Wizard-of-Oz



Example: Aardvark startup - acquired by Google and subsequently abandoned/shutdown by Google

Prototyping

Low-Fidelity Prototyping: Wizard-of-Oz

Some of the potential ethical issues associated with this approach include (1/2):

- Deception: Misleading participants about automated system interaction raises ethical concerns and impacts trust.
- Informed Consent: Participants must be fully informed about the prototyping method, including the human operator, and provide consent.
- Privacy and Data Protection: Protecting participant information and ensuring confidentiality is essential. Designers must handle sensitive data responsibly and adhere to privacy regulations.

Prototyping

Low-Fidelity Prototyping: Wizard-of-Oz

Some of the potential ethical issues associated with this approach include (2/2):

- Emotional Well-being: Participants may have emotional reactions upon discovering they interacted with a human instead of a real system. Designers should monitor and support participants to address any negative emotions.
- Psychological Impact: Consider potential psychological impacts and ensure participant well-being during Wizard-of-Oz prototyping.

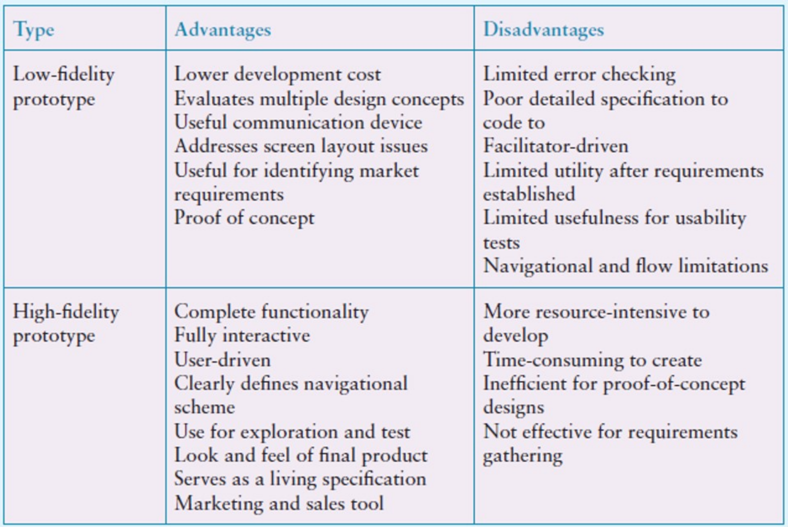
Prototyping

High-Fidelity Prototyping

- Utilizes materials resembling the final product
- Resembles the final system more closely than low-fidelity prototypes
- Involves integration of existing hardware and software components
- Caution: Participants may perceive it as a complete system, leading to potential compromises.

Prototyping

Low-Fidelity vs High-Fidelity Prototyping



Type	Advantages	Disadvantages
Low-fidelity prototype	Lower development cost Evaluates multiple design concepts Useful communication device Addresses screen layout issues Useful for identifying market requirements Proof of concept	Limited error checking Poor detailed specification to code to Facilitator-driven Limited utility after requirements established Limited usefulness for usability tests Navigational and flow limitations
High-fidelity prototype	Complete functionality Fully interactive User-driven Clearly defines navigational scheme Use for exploration and test Look and feel of final product Serves as a living specification Marketing and sales tool	More resource-intensive to develop Time-consuming to create Inefficient for proof-of-concept designs Not effective for requirements gathering

Prototyping

Compromises in Prototyping

- Prototyping entails compromises in various aspects.
- Software-based prototypes may have compromises such as slow response, sketchy icons, or limited functionality.
- In-the-wild prototypes may be operational but not fully robust.
- Two common types of compromise are horizontal, offering a wide range of functions with little detail, and vertical, providing detailed functionality for only a few functions.
- Another common compromise is between robustness and changeability.
- Caution is advised as prototypes should not be mistaken for the final engineered product, considering the compromises made.

Prototyping

Concrete Design: Enhancing User Experience (1/2)

- By focusing on concrete design, designers can enhance the user experience by refining visual elements, considering user characteristics and context, ensuring accessibility, accommodating localization needs, and incorporating diverse cultural perspectives.
- This approach leads to more inclusive, engaging, and culturally sensitive designs.

Concrete Design: Enhancing User Experience (2/2)

- Concrete design focuses on refining and specifying design elements - color, icons, buttons, and interaction devices.
- Design for inclusiveness by accommodating diverse user characteristics - consider input and output modes that suit user preferences and capabilities.
- Use Web Content Accessibility Guidelines to ensure inclusivity.
- Tailor the design to support different languages and cultural contexts -consider adapting navigation, icons, and metaphors for global users.
- Integrate indigenous knowledge into the design process - how design elements can reflect diverse cultural perspectives.

Prototyping

Generating Prototypes

- By generating prototypes, designers can gain insights into user interactions, identify design considerations, and foster a collaborative design process.
- Prototyping serves as a valuable tool in the iterative design cycle, aiding in creating user-centered and impactful solutions.
- Benefits of Prototyping:
  - Visualizes the user's journey and interaction with the product.
  - Promotes early exploration of design issues and considerations.
  - Facilitates effective communication and collaboration among team members.
  - Enables iterative design and feedback to refine the user experience.

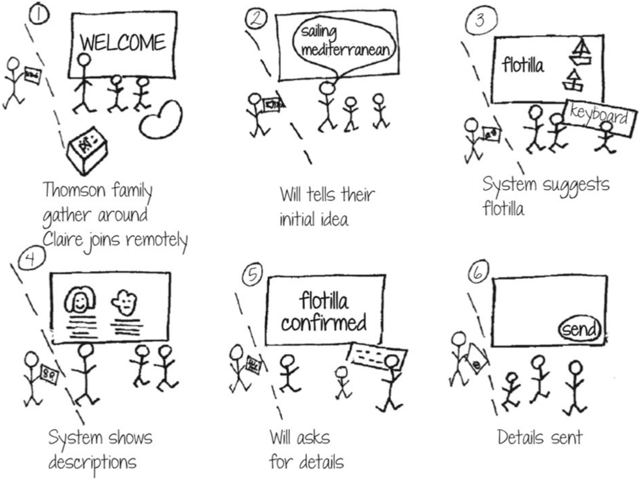
Prototyping

Generating Prototypes: From Storyboards to Card-Based Prototypes

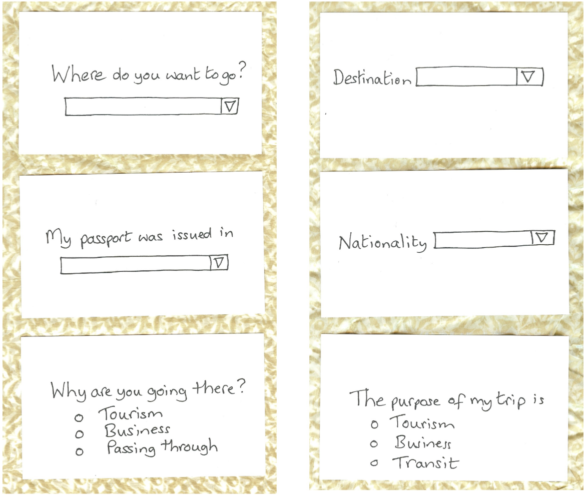
- Storyboard Generation:
    - Break down scenarios into sequential steps.
    - Create a scene for each step to visualize the user's interaction.
    - Storyboarding prompts designers to consider design issues effectively.
- Card-Based Prototype Generation:
    - Utilize storyboards or use cases as a basis for creating a card-based prototype.
    - For each step in the use case, identify the required interaction elements.
    - Draw a card that captures the specific interaction element for that step.

Prototyping

Generating Storyboards



Generating Card-Based Prototypes



Where do you want to go?

My passport was issued in

Why are you going there?
- ○ Tourism
- ○ Business
- ○ Passing through

Destination

Nationality

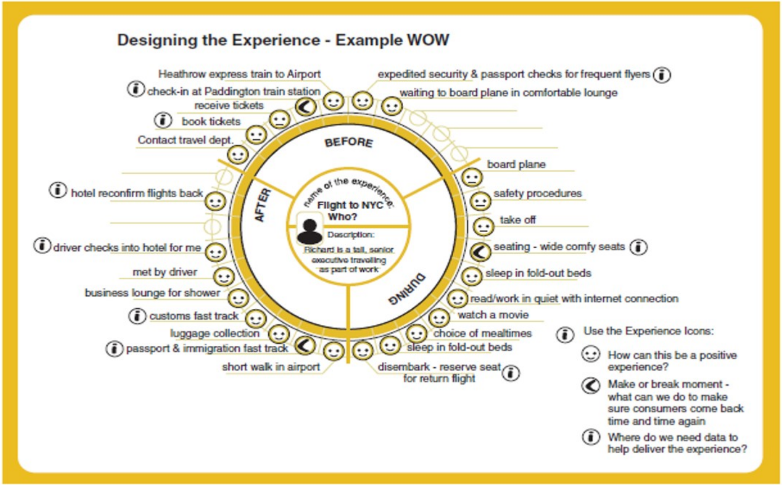
The purpose of my trip is
- ○ Tourism
- ○ Business
- ○ Transit

Prototyping

Mapping the Overall Experience - Not examinable

- By utilizing design maps, customer journey maps, experience maps, and user flows, designers can gain valuable insights into the overall user experience.
- These visual representations aid in identifying key touchpoints, understanding user needs, and informing design decisions to create meaningful and engaging experiences.
- Benefits of Mapping the Overall Experience:
  - Provides a holistic view of the user's journey and interactions.
  - Identifies pain points, opportunities, and areas for improvement.
  - Enhances empathy and understanding of the user's perspective.
  - Facilitates collaboration and communication among stakeholders.

Mapping the Overall Experience: An Experience Map as a Wheel - Not examinable



Designing the Experience - Example WOW

Heathrow express train to Airport
check-in at Paddington train station
receive tickets
book tickets
Contact travel dept.
hotel reconfirm flights back
driver checks into hotel for me
met by driver
business lounge for shower
customs fast track
luggage collection
passport & immigration fast track
short walk in airport
expedited security & passport checks for frequent flyers
waiting to board plane in comfortable lounge
board plane
safety procedures
take off
seating - wide comfy seats
sleep in fold-out beds
read/work in quiet with internet connection
watch a movie
choice of mealtimes
sleep in fold-out beds
disembark - reserve seat for return flight
BEFORE
AFTER
DURING
name of the experience
Flight to NYC
Who?
Description:
Richard is a tall, senior executive travelling as part of work

Use the Experience Icons:

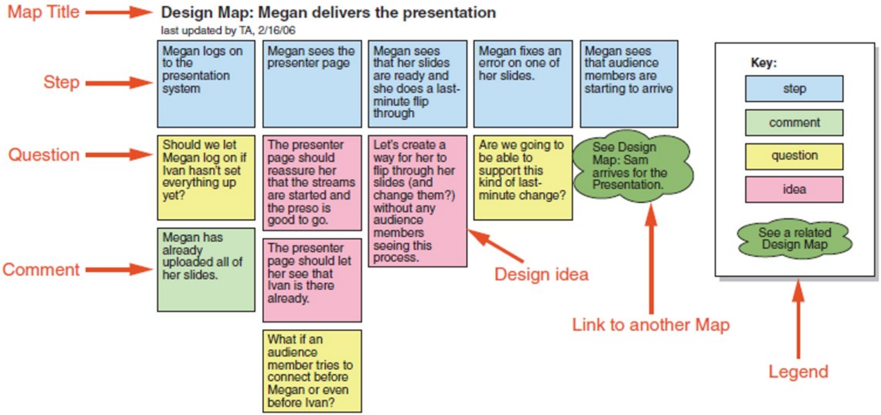
☺ How can this be a positive experience?

Make or break moment - what can we do to make sure consumers come back time and time again

ⓘ Where do we need data to help deliver the experience?

Prototyping

Mapping the Overall Experience: An Experience Map as a Timeline- Not examinable



Design Map: Megan delivers the presentation

last updated by TA, 2/16/06

Megan logs on to the presentation system

Megan sees the presenter page

Megan sees that her slides are ready and she does a last-minute flip through

Megan fixes an error on one of her slides.

Megan sees that audience members are starting to arrive

Should we let Megan log on if Ivan hasn't set everything up yet?

Megan has already uploaded all of her slides.

The presenter page should reassure her that the streams are started and the preso is good to go.

The presenter page should let her see that Ivan is there already.

What if an audience member tries to connect before Megan or even before Ivan?

Let's create a way for her to flip through her slides (and change them?) without any audience members seeing this process.

Are we going to be able to support this kind of last-minute change?

See Design Map: Sam arrives for the Presentation.

Prototyping

Construction: Physical Computing

- By utilizing toolkits such as Arduino, LilyPad, Raspberry Pi, BBC micro:bit, and MaKey MaKey, designers can bring their ideas to life through physical computing.
- These platforms provide a foundation for building interactive prototypes, fostering creativity, and enabling innovative user experiences.
- Benefits of Physical Computing:
  - Enables the integration of physical components and interactivity into prototypes.
  - Facilitates the exploration of tangible and interactive user experiences.
  - Encourages interdisciplinary collaboration between designers, developers, and makers.
  - Supports rapid prototyping and iterative design processes.

Prototyping

The Four Major Software Prototyping Techniques

- When it comes to software prototyping, there are four prominent methods to consider.
- These techniques can greatly simplify the process of creating improved and presentable software prototypes.
  1. Incremental Prototyping
  2. Throwaway Prototyping
  3. Extreme Prototyping
  4. Evolutionary Prototyping
- While there are other methods available, these four techniques stand out as the most effective options for creating enhanced and presentable software prototypes.
- Each method offers a slightly different approach, but all contribute to the overall goal of building superior software

The Four Major Software Prototyping Techniques - Incremental Prototyping: (1/2)

- Incremental prototyping is a software development approach that involves creating iterative versions of a prototype, with each version being an improvement over the previous one.
- This method follows a linear development process, where features are added, design is enhanced, and bugs are fixed at each stage.
- By following incremental prototyping, you can gradually enhance your software prototype and ensure a smooth development process.
- Each iteration brings you closer to a more refined and reliable end product.

Prototyping

The Four Major Software Prototyping Techniques - Incremental Prototyping (2/2)

Benefits of Incremental Prototyping:

- Easy and quick creation of software prototypes.
- Allows for frequent testing and validation within a short timeframe.
- Ensures continuous improvement and refinement of the prototype.
- Minimizes errors and reduces the risk of failure in the final software.

Note: Incremental Prototyping focuses on Building Better Software Prototypes

Prototyping

The Four Major Software Prototyping Techniques - Throwaway (Rapid) Prototyping (1/2)

- Throwaway prototyping, also known as close-ended prototyping, is a method where the main goal is to refine a specific aspect of the prototype.
- It is not intended for integration into the final software product - used to gather user feedback and make improvements based on that feedback.
- With throwaway prototyping, you can efficiently collect valuable insights from users and make informed decisions about the next steps in the development process.
- By focusing on specific aspects, you can create a more effective and user-centered prototype.

Prototyping

The Four Major Software Prototyping Techniques - Throwaway (Rapid) Prototyping (2/2)

Advantages of Throwaway Prototyping

- Allows for targeted refinement of specific prototype features.
- Feedback-driven approach for iterative improvements.
- Reusable components can be retained for future use.
- Saves time and effort by eliminating the need for extensive documentation.

Note: Throwaway Prototyping Focuses on Specific Aspects

Prototyping

The Four Major Software Prototyping Techniques - Extreme Prototyping (1/2)

- Extreme prototyping is a valuable approach primarily used for developing web applications, consisting of three stages:
  - Building a Static Prototype: Creating HTML pages that represent the user interface and overall structure of the application.
  - Programming Screens with a Stimulated Service Layer: Adding functionality to the screens by simulating the service layer that interacts with the backend systems.
  - Implementing the Services: Integrating the actual services and backend functionality into the prototype.
- By adopting extreme prototyping, developers can mitigate risks, optimize performance, and create web applications that effectively meet user needs and expectations.

Prototyping

The Four Major Software Prototyping Techniques - Extreme Prototyping (2/2)

Benefits of Extreme Prototyping:

- Early Bug Detection: Identifying and addressing bugs and issues before the application reaches the production stage, resulting in cost savings and improved efficiency.
- Enhanced User Engagement: Prioritizing user experience and performance at each stage to meet user expectations and increase engagement.
- Iterative Development: Iteratively refining the prototype to ensure it evolves into a final product that effectively engages users.

Prototyping

The Four Major Software Prototyping Techniques - Evolutionary Prototyping (1/2)

- Evolutionary prototyping, also known as breadboard prototyping, is a fundamental and developer-focused approach to prototyping.
- This method involves creating a robust prototype that incorporates a comprehensive set of features and functionalities based on well-understood requirements.
- As the process evolves, additional enhancements and improvements are integrated, resulting in a refined prototype.
- With evolutionary prototyping, developers can build robust prototypes that evolve alongside the project, resulting in a refined and effective end product.

The Four Major Software Prototyping Techniques - Evolutionary Prototyping (2/2)

Benefits of Evolutionary Prototyping:

- Improved Understanding: Developers gain understanding of the project requirements and refine the prototype accordingly.
- Iterative Refinement: The iterative nature of this method allows for ongoing improvements and adjustments to align the prototype with user expectations.
- Risk Mitigation: Early identification of potential issues enables proactive problem-solving and risk mitigation.

Note: Evolutionary Prototyping Focuses on Building Robust and Refined Prototypes

Prototyping

Reading Assignment: The Windws$^{(R)}$ 95 User Interface Prototyping and Usability Testing

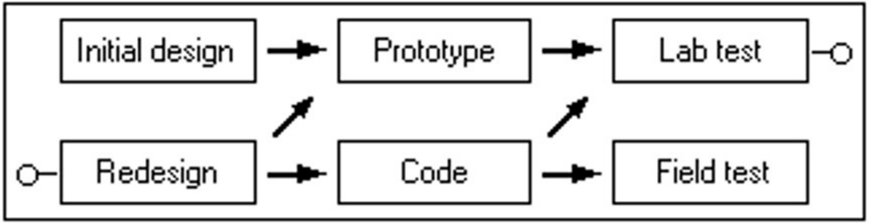


Figure 1: Windows 95 Iterative Design Process