

UX Design and Software Engineering

SWEN444

Why Study Human-Computer Interfaces (HCI) Design?

- Over the last two decades **UX design** has **matured**
- Many (most?) designs are pretty good (or are they?)
- **Design and implementation** have been **commoditized**, especially for **web app** and **mobile app** platforms
 - Best practices checklists
 - Pattern libraries
 - Style guidelines
 - Code libraries
 - Design tools

Because...

- You still have to make “good” **design decisions** based on understanding **user goals and requirements**
- You may face **specialized design problems**, e.g., embedded and/or IoT devices
- You will need to **accommodate evolving technologies** such as **AI**, data **visualization**, and **non-traditional technologies**.

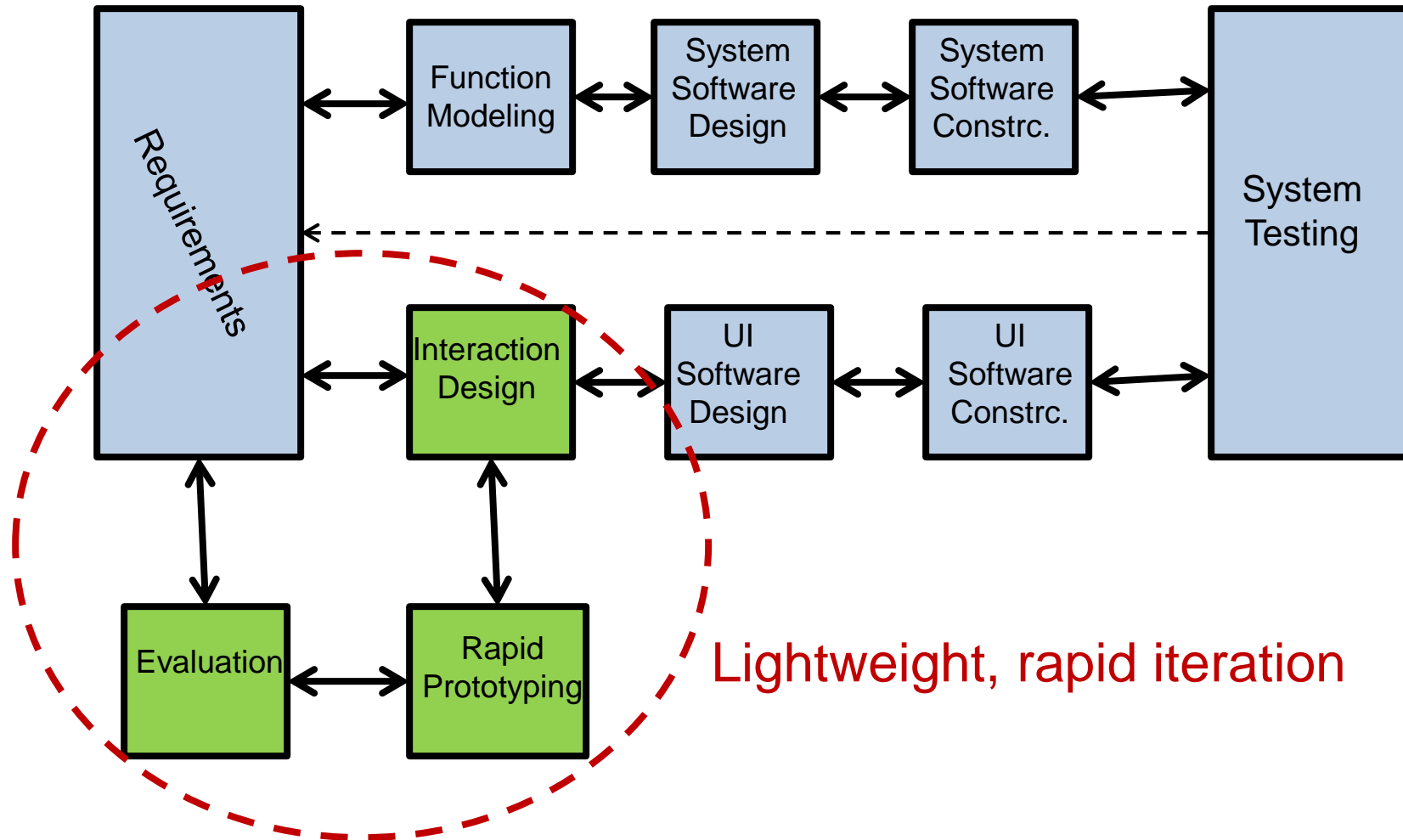
“Human Centered Requirements and Design”

Learning Objectives

- Primary – **design thinking** methods, life cycle processes, and techniques to create a HCI design
- Secondary – design visual aesthetics
 - This is NOT about graphical design!

NOTE: Our focus will be on traditional graphical user interfaces (GUI)

UX Design and the Software Process



Two Distinct Roles

- Interaction **UX designer** and **UI software designer**
 - Note: sometimes the software engineer fills both roles!
- Premise – **user focus** and **interactive design expertise** should result in a **better UX**
- Inherent **conflict of interest**: What's best for the user is seldom easiest for the software developer!

UX and Software Architecture Design

- **Usability** is a **quality attribute** requirement
- **Usability scenarios drive software architecture design;** examples ...
 - Cancel task
 - Undo/redo action
 - Task progression reporting
 - Error recovery
 - Internationalization
- Design patterns – e.g., MVC and others

SWEN444 Vs SWEN440

- What's the same?
 - Discover requirements
 - Analyze requirements
 - Design a solution
- What's different?
 - 444
 - Design the HCI
 - Focus on **users** and their UX related interaction goals and requirements
 - 440
 - Design the software architecture for the **system**
 - Consider all stakeholders
 - System functional and non-functional requirements
 - Methods and techniques (limited overlap)