Environmental Decision Support Systems: A Human Factors Perspective

Prof. Dr. Steven P. Frysinger

James Madison University
Harrisonburg, Virginia USA

and

cismet GmbH

EDSS

- Decision Support Systems attempt to facilitate a “natural intelligence” approach to human problem solving
- EDSS specifically focuses on environmental problems - a very large problem space

EDSS Technology Integration

- Environmental Decision Support Systems typically integrate a wide range of technologies
  - Geographic Information Systems
  - Mathematical Process Models
  - Monte Carlo Simulation
  - Expert Systems
  - Linear Programming Optimization
  - …

EDSS

- Environmental decision makers WILL make decisions with or without the benefit of science
- EDSS is intended to bring science to these decision makers
- But historically, EDSS has had limited success in spite of considerable effort in the last 25 years
  - Decision makers still frequently work “in the dark”

Why?

The Problem

- EDSS is fundamentally about people…

  …and how they make decisions
- Yet in EDSS development human factors are usually only considered in *ad hoc* ways
- We need to explicitly address the application of human factors engineering to the design and development of environmental decision support systems

The Nature of Environmental Decisions

- Environmental decisions are particularly complex in nature:
  - Spatial
  - Temporal
  - Uncertain
  - Risk-oriented
- These characteristics make them especially difficult for humans to approach

Environmental Decision Makers

- Environmental decisions are made by an enormous variety of individuals
  - Scientists and engineers
  - Business people
  - Community leaders
  - Voters (perhaps the most important decision makers)
- The degree of technical training varies widely
- To support these people, the EDSS must acknowledge and accommodate this variety

Human Factors Methods

- Traditionally, attention is primarily paid to Interface Design
  - GUI, buttons, colors, &c
- We need to focus on “Interaction Design”
  - Consider the total relationship to the human
    - Needs, goals, and objectives
    - Conceptual Model
- Rigorous evaluation of the interaction design is at least as important as testing of software

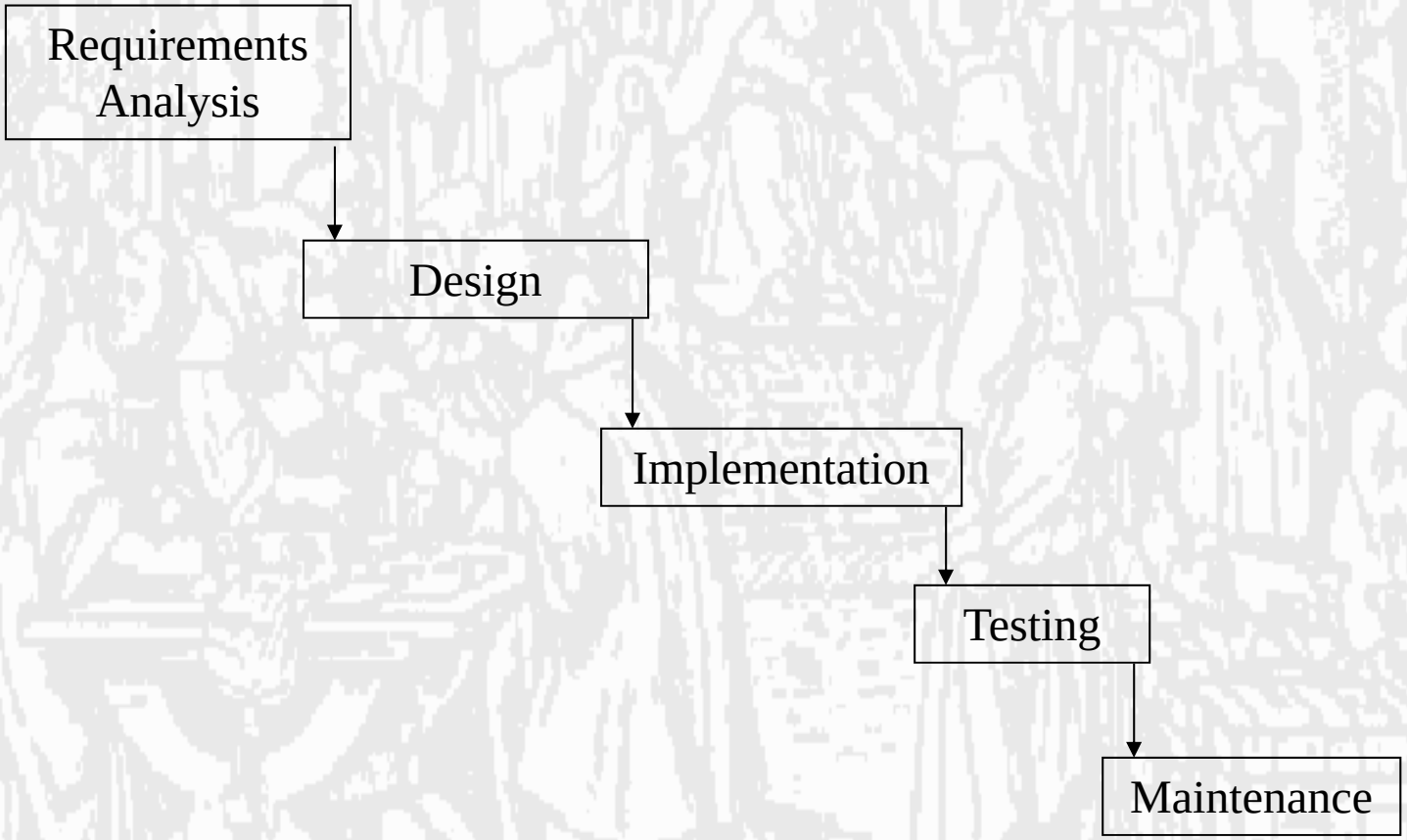
Interaction Design

- Identify the Stakeholders
  - Customers, “innocent bystanders”, users
- Identify the Users
  - Primary: Frequent hands-on users of the system
  - Secondary: Occasional users, or who use the system through an intermediary
  - Tertiary: Those who are affected by the system without having direct or indirect interaction with it
- Engage in user-centered software engineering
  - The humans are part of the system!

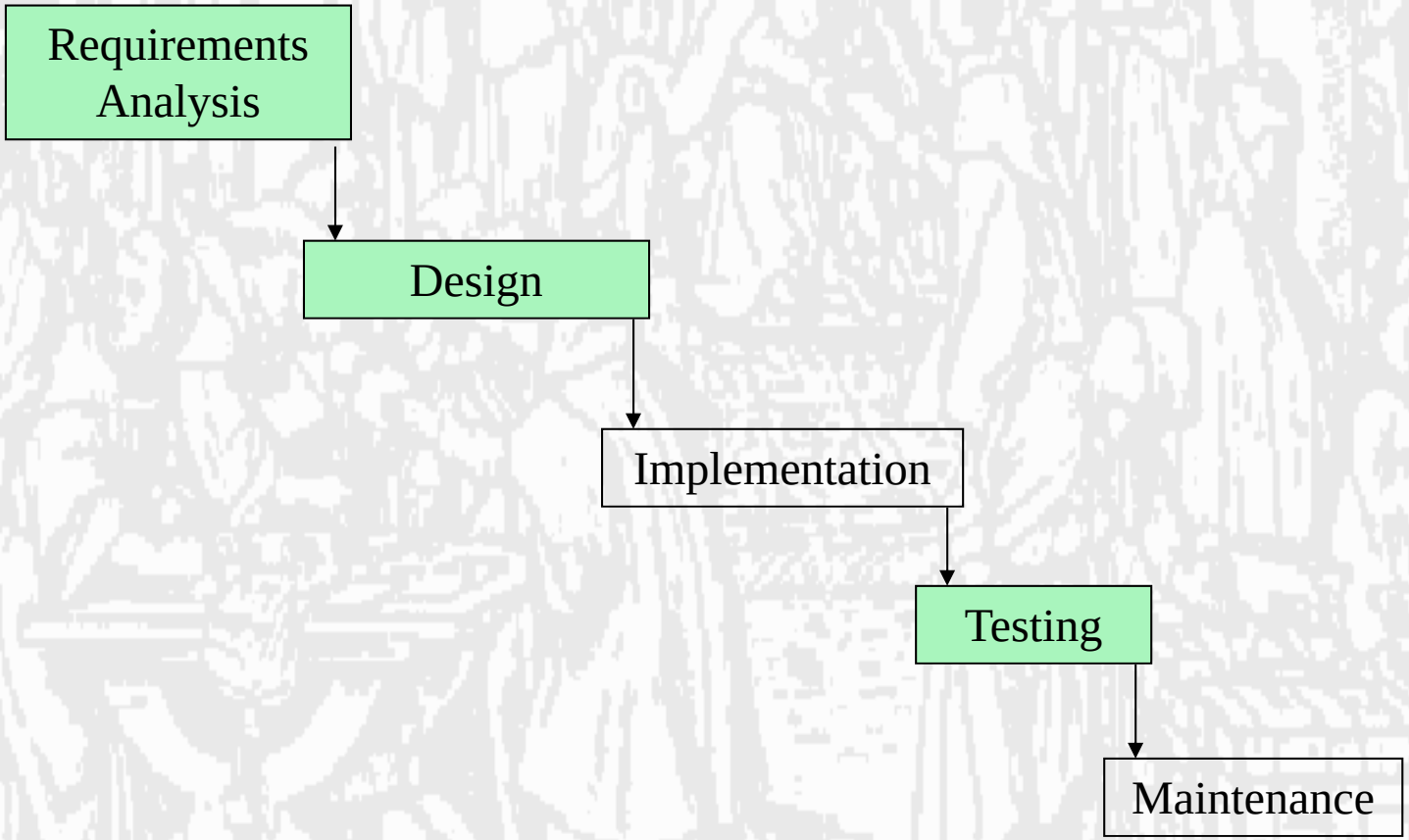
The Interaction Design Process

- Five Basic Activities:
  - Identifying needs and establishing requirements
  - Exploring alternative designs meeting requirements
  - Building interactive prototypes for assessment
  - Continuous evaluation throughout development
  - Quantitative and qualitative testing before delivery
- Notes:
  - Users should be involved throughout the process
  - Specific usability goals should be agreed at the outset
  - Iteration is inevitable

HCI and the Software Lifecycle



HCI and the Software Lifecycle



Requirements Analysis

- Needs Assessment
  - People have difficulty knowing what they need until they know what is possible
  - It is natural for designers to design “for themselves”, based on what they would like – this is a frequent cause of system failure
- Task Analysis
  - A structured mechanism for breaking down the current or projected work of the user

System Design and the Interactive User Interface

- Interface design can occur in the software design phase, but is better accomplished in the requirements phase
- Prototyping is a critical element in the design process
  - Low fidelity vs. high fidelity
- “Conceptual Design” transforms the user requirements into a conceptual model:
  - a description of the proposed system in terms of a set of integrated ideas and concepts about what it should do and how it should appear.

Functional Testing and Human Performance Experiments

- Software engineers wouldn't release software without testing, but the human interaction design is rarely tested
  - Lack of training/know-how is the impediment
- Techniques from experimental psychology allow both qualitative and quantitative evaluation to be carried out throughout the lifecycle

Implications of Human Factors Engineering for EDSS

- The concept of a “generic” EDSS is fundamentally flawed:
  - Too many different types of users
  - Too many different types of decisions
- “One size fits all” in this case really means it doesn’t fit anyone very well:

  Generic systems are too complex
- An open and adjustable architecture is a possible compromise

Unsolicited Advice

- Identify ALL stakeholders and users
- Engage users in the interaction design process
- Integrate human factors into the total lifecycle
- Task analysis and evaluation are at least as important as GUI design
- Evaluation is more than asking the user if they like the system – design the experiments

Questions?