# CONSTRUCTION, EVOLUTION AND PROTOTYPING

SU:E16:L2

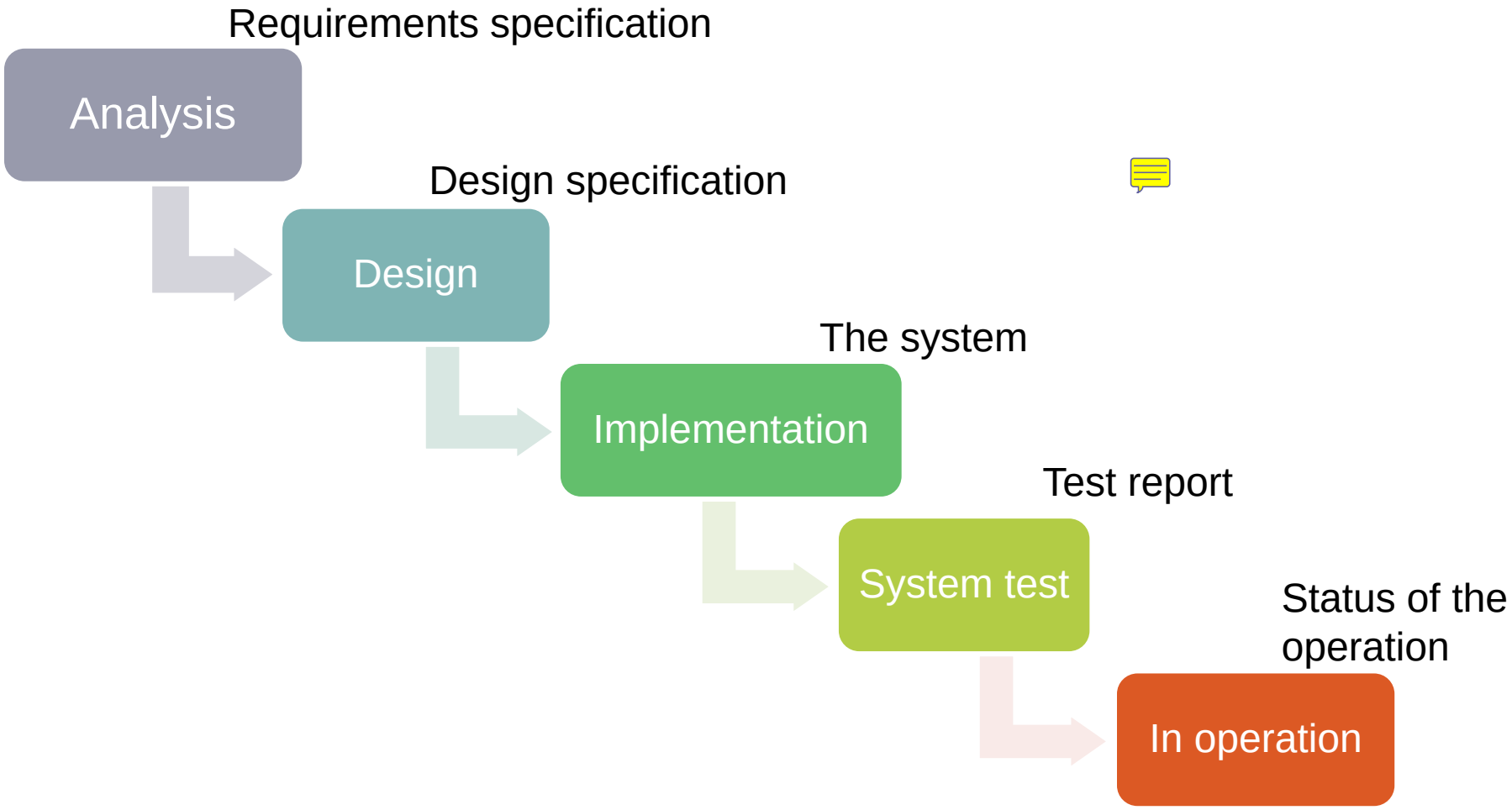
# CONSTRUCTION AND EVOLUTION

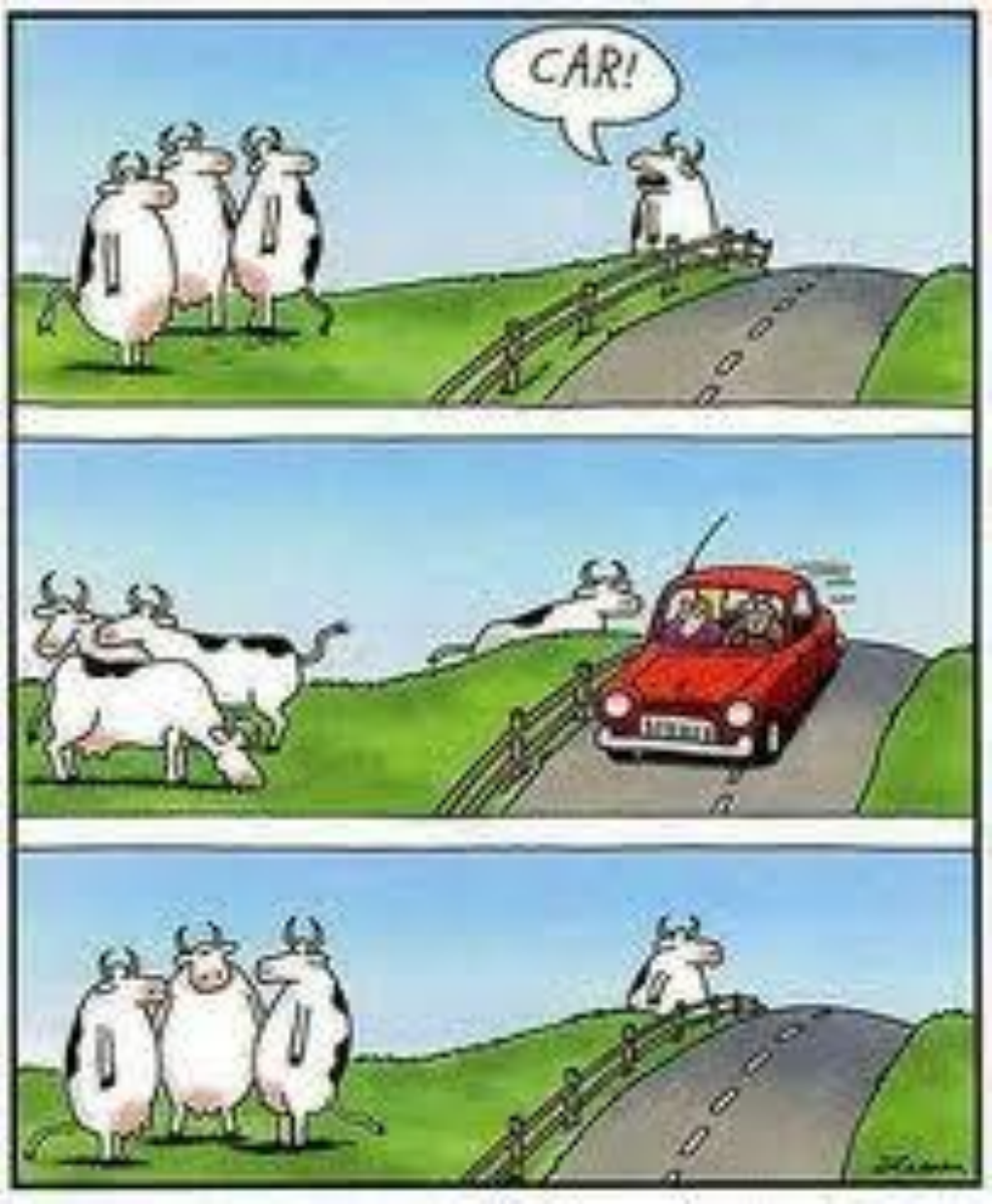
# CONSTRUCTION

- **A bureaucratic approach to systems development**
- **Relies on overall plans**
  - Concrete implementations of the general guidelines
- **Mathematical problem solving**
- **Users play a passive role**
  - Provide information, approve decisions
- **Linear approach**

- **Works well for stable and explicitly stated problems**
- **Inability to respond efficiently to change**

# CONSTRUCTION: THE WATERFALL APPROACH



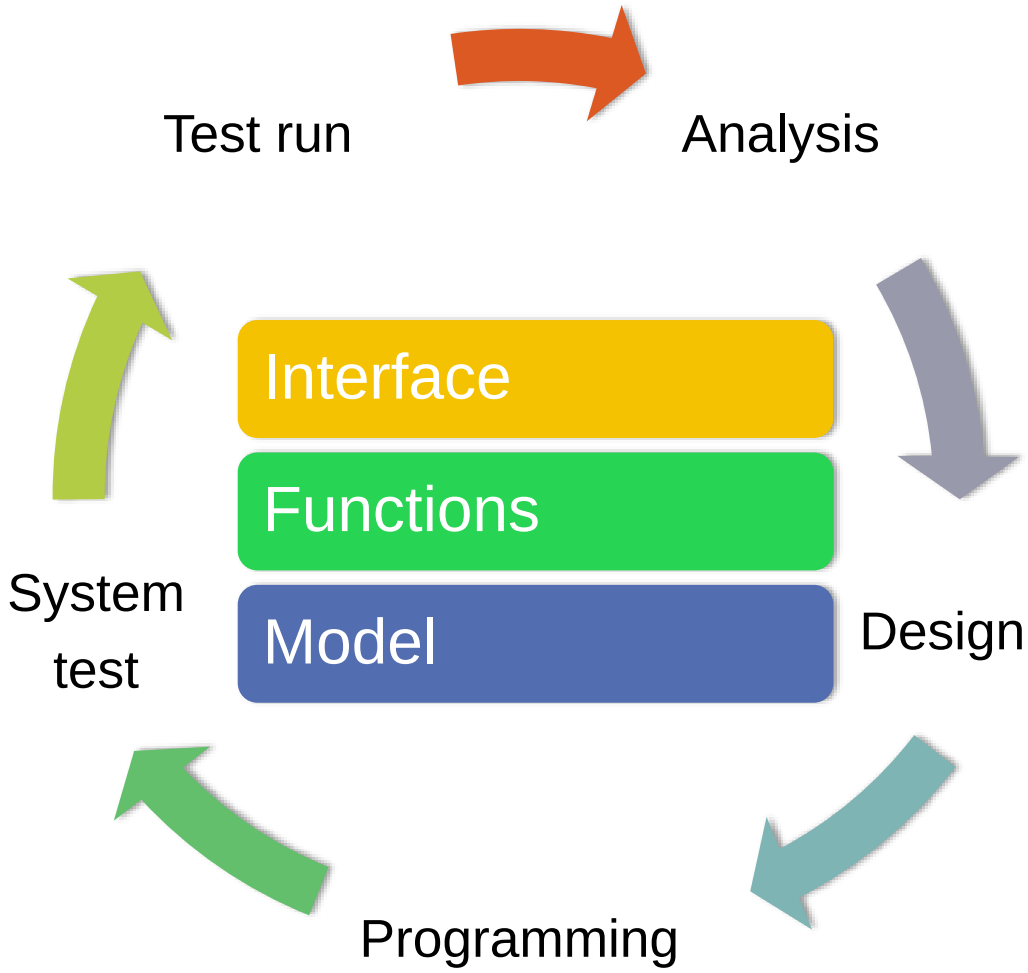

CAR!

# EVOLUTION

- **Real problems are rarely clear and precise**
  - Planning is a risky business
  - Trial and error reduce uncertainty
- **Evolution recognizes and emphasizes the uncertainties**
  - The problem is interpreted and restated (iterative approach)
  - The result is a *satisfactory* version of the system
  - The system may be developed further
- **Close communication and interactions with users**

- **Works well in changing environments**
- **Harder to know when the project is done**

# EVOLUTION: THE ITERATIVE APPROACH



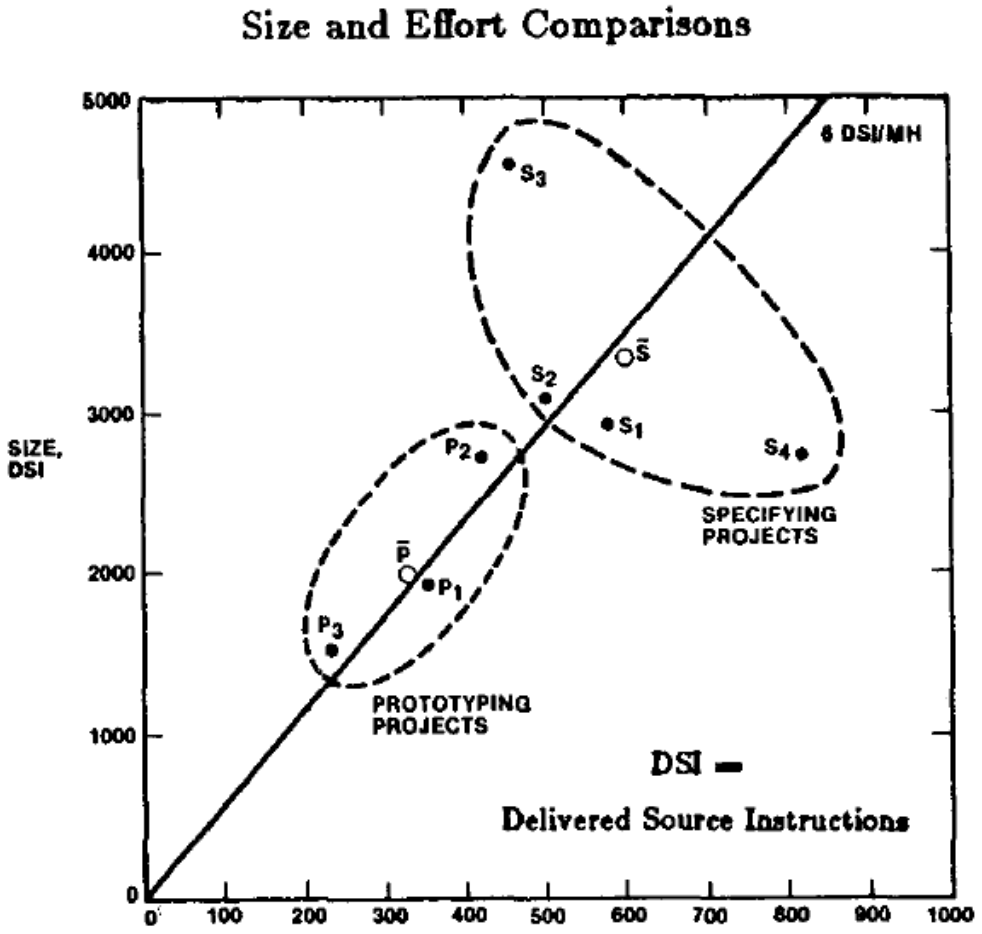
# BOEHMS EXPERIMENT (1)

**7 groups solve the same set assignment:**

- **develop a system that can be used by project managers for estimating resources**

**4 groups worked specification-oriented**

**3 groups worked with prototypes**



# BOEHMS EXPERIMENT (2)

**All 7 products were evaluated from the same criteria**

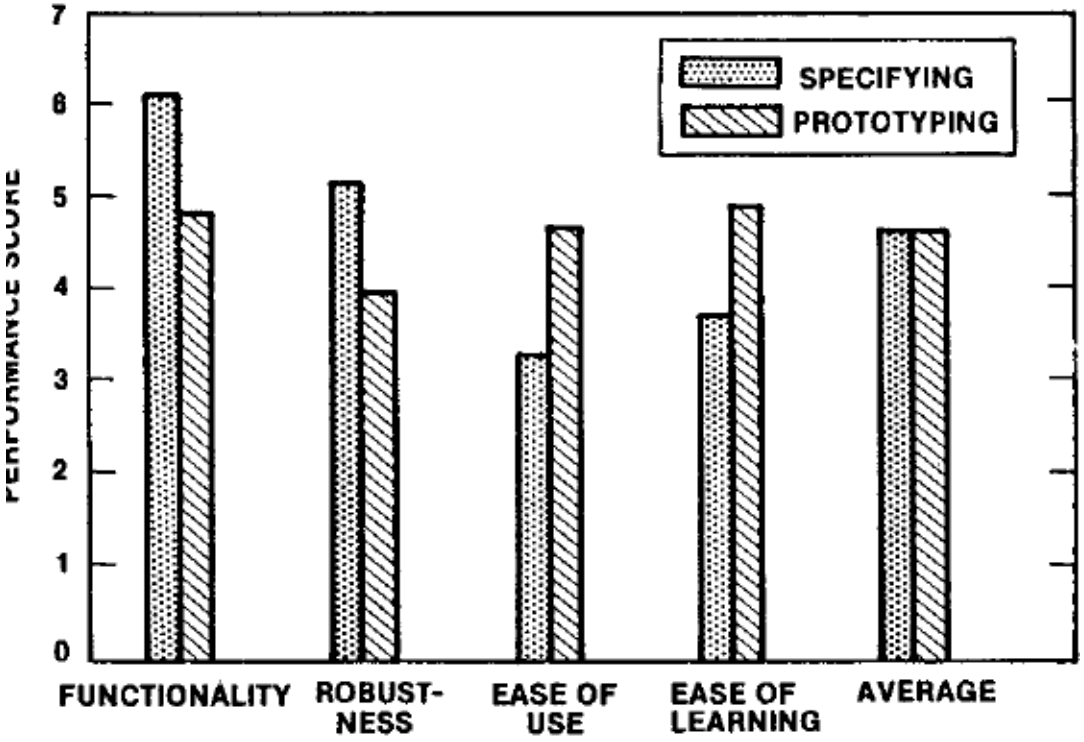
**Prototyping seemed advantageous**

- half the code
- easy to use and learn
- better interfaces
- a functioning system early

**The advantages were not free**

- Less effort on planning - more effort on fixing and testing
- Lack of specifications (difficult integration)
- Design was less coherent
- Rated lower in overall functionality and robustness

Figure 2. Specifying vs. Prototyping: Performance Comparisons



# GROUP DISCUSSION

- **Discuss the advantages and disadvantages of construction and evolution respectively!**

# TWO WORLD-VIEWS

## Mechanistic

- Algorithmic (Church-Turing)
- Rational manipulation of symbols (Descartes)
- “Machines can think – when we have completed the programs”
- Systems development is **construction**

## Romantic

- Competences and knowledge is richer than information and data
- Information = data + interpretation
- “Machines will never think”
- Systems development is **evolution**

# COMPLEXITY AND UNCERTAINTY

## Complexity

- A lot of information
- ”complicated” but defined problem
- Great solution frame
  - 8 queens problem,
  - route planning,
  - room allocation,
  - sorting
- **Strategy:**
  - Categorize and abstract
  - Divide into sub problems until each sub problem has a simple solution

➢ **Construction**

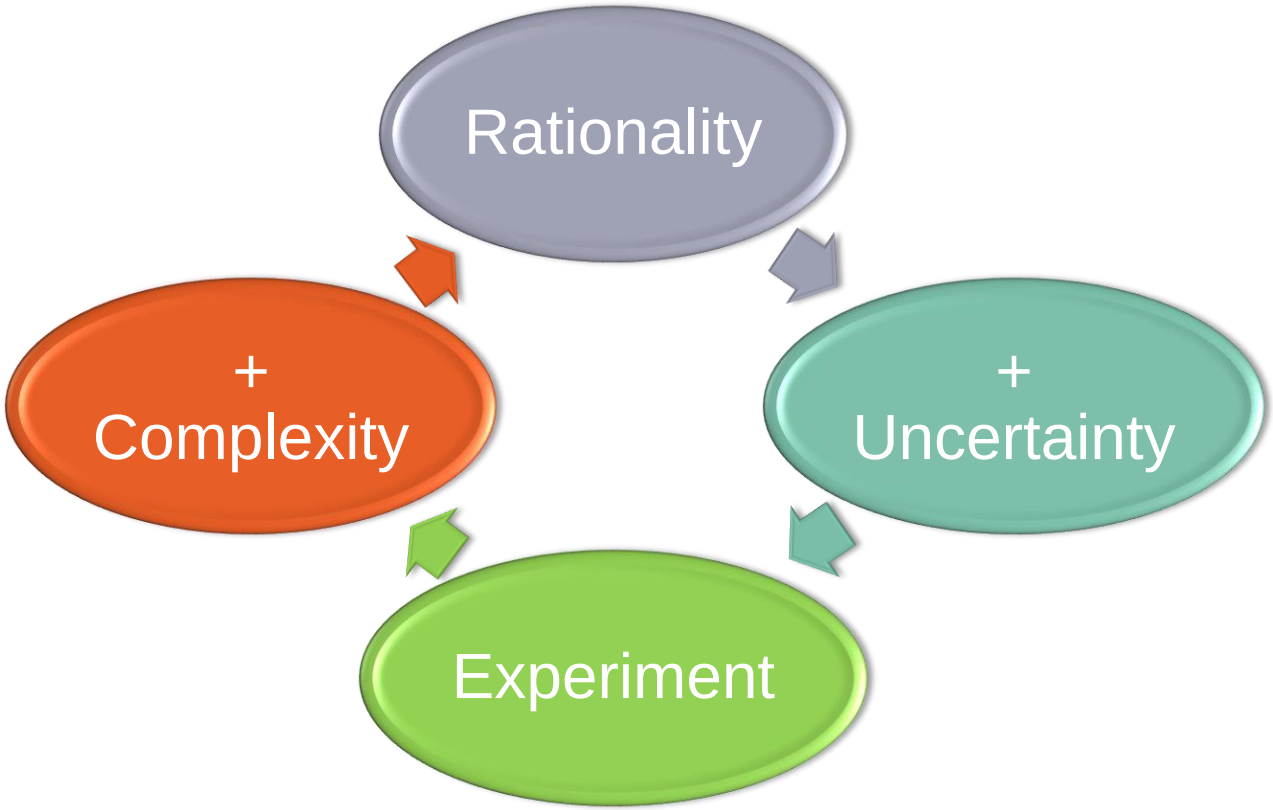
## Uncertainty

- Insufficient or ”unreliable” information about the problem and its solution
- The requirements for the system is unknown or only partly described
- **Strategy:**
  - Collect more (and more reliable) information about the problem and possible solutions
  - Experiment with different solutions

➢ **Evolution**

**What if both complexity and uncertainty is high?**

# THE PRINCIPLE OF LIMITED REDUCTION




If you reduce the uncertainty the complexity is hightened
If you reduce the complexity the uncertainty is hightened

3

# PROTOTYPING PRINCIPLES

# CONCEPTS

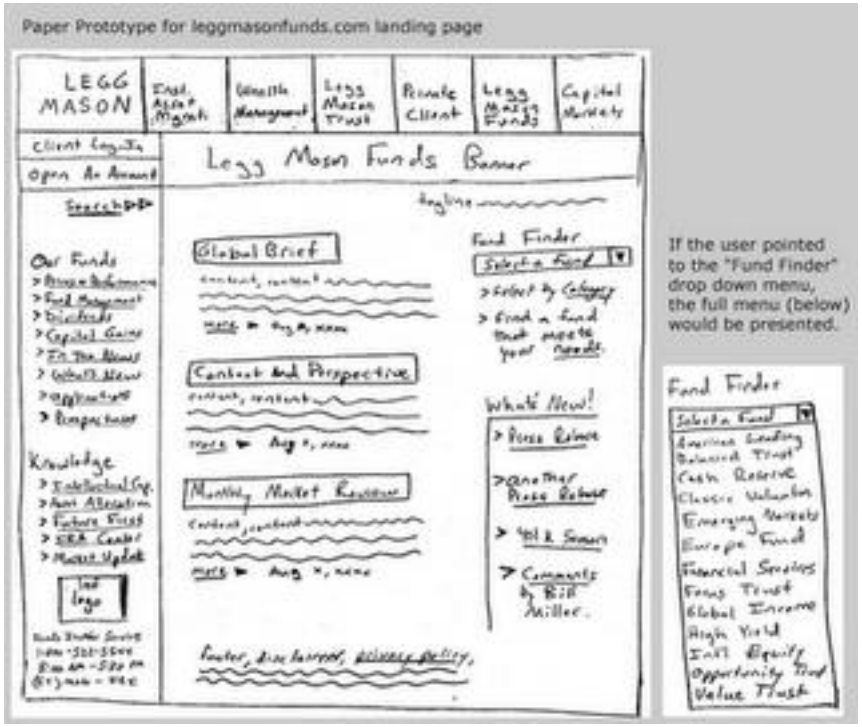
- **Client:**
  - Initiates, makes agreements and signs the contract with a software manufacturer.
- **Users:**
  - Work with or uses the application system (may not just be people looking at a screen or using a mouse).
- **Software manufacturer**:
  - Is responsible for developing and delivering the application system (may not just be the software system itself).
- **Prototype:**
  - An early version of the future system;
  - Operational;
  - Communication medium between users and developers

# PROTOTYPE CLASSIFICATION

| | |
|---|---|
| Presentation prototype | • Developed quickly using few resources<br>• Early<br>• Determination of overall requirements<br>• A first impression |
| Prototype proper | • Provisional operational software system<br>• Illustrate specific functions<br>• Reveals design problems<br>• Realistic |
| Pilot system | • Taken into use<br>• Not the final system |

# PAPER PROTOTYPE

- **Hanmail Paper prototype on YouTube**




- What can such a test be used for?
- How would you plan such a test?

# POWERPOINT PROTOTYPE

Wikipedia

Random article

**Wacho** (or **Waccho**) (probably Waldchis) was king of the Lombards before they entered Italy from an unknown date (perhaps circa 510) until his death in 539. His father was Unichis. Wacho usurped the throne by assassinating (or having assassinated) his uncle, King Tato (again, probably around 510). Tato's son Ildchis fought with him and fled to the Gepids where he died.[1] Wacho had good relations with the Franks. Wacho married three times. His first marriage was to Radegund, daughter of Bisinus, King of the Thuringi. His second marriage was to Austrigusa, a Gepid possibly named after her maternal descent from Ostrogothic rulers.

# PP PROTOTYPE INSTRUCTIONS

- Genap – example

- Toolkit – example
  - Get the example from Moodle
  - Test and evaluate the prototype

- How to – example
  - See the video during the exercises

# PURPOSES OF PROTOTYPING

**Exploratory prototyping**

- Identify problems and needs
- Gain insights
- Discussions may be around paper prototypes or proper prototypes

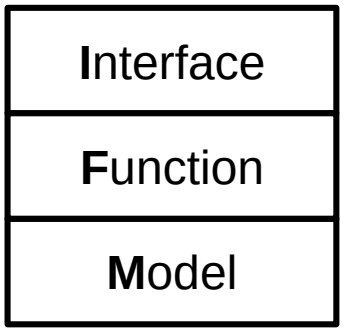
**Experimental prototyping**

- Test and refine design ideas
- Choose between alternatives
- Prototypes proper help clarify

**Evolutionary prototyping**

- A continuous process
- The system is developed incrementally
- Pilot system

# HORIZONTAL & VERTICAL PROTOTYPING




## Horizontal prototyping

- The top layers of the system
- All of I,
- Some F,
- No M

## Vertical prototyping

- A part of the system is implemented completely
- Some I,
- Some F,
- Some M

Every prototype can be thrown away

# PRINCIPLES

- Specifying requirements is a difficult task

- A prototype is part of the application system specification
- Prototypes are a step on the way (increments) to the future system
- Prototypes are used exploratorily or to test solutions

# COOPERATIVE INTERACTION (1)

- **Being inside – the adaptive computer**
  - In wizard-of-oz prototyping, the wizard is the computer
  - Possible to analyze user input and adjust in the moment

- **The computer personified**
  - Errors are not blamed on the computer
  - Room for reflections on the requirements

# COOPERATIVE INTERACTION (2)

- **Rapid iterative test of design ideas**

  - Easy to make modifications to the prototype
  - Valuable

- **Collaboration and dialogue – user involvement**
  - Users are interested and enthusiastic
  - Basis for constructive dialogues

- **Methodology and tool problems**
  - Incorrect wizard behavior
  - The user becomes aware of the wizard
  - User bias – ‘emotional stakeholders’

# EXPERIMENT WITH PROTOTYPES

1. **Planning**
   - Describe the prototype content
2. **Development**
   - Start with simple prototypes on paper
   - Simple prototypes in for example Power point
   - Functioning prototypes
3. **Preparation**
   - Cooperation
   - Realism
   - Which users

4. **Test**
   - Documentation
5. **Summarizing**

# PLANNING A PROTOTYPE TEST

Conference planning chapter 19

| | Focus | Limitations | Prerequisites |
|---|---|---|---|
| Interface | Only one screen picture related to participants | Full layout of participant picture with all keys and fields included | All keys give informative answers |
| Function | Updating participant data | No other functions | Registration of participants should work |
| Model | | | Should hold at least 10 participants and their payments |

# GROUP DISCUSSION

## ASSIGNMENT

- Plan a horizontal and a vertical prototype test of the official login (NemID) to determine requirements for the system


## CONCEPTS

- Horizontal prototype
- Vertical prototype
- Clients, users, developers
- Presentation prototype
- Prototype proper
- Pilot prototype