



## ***Software Life Cycle Models***



## ***Software Life Cycle Models***

◇ Waterfall model

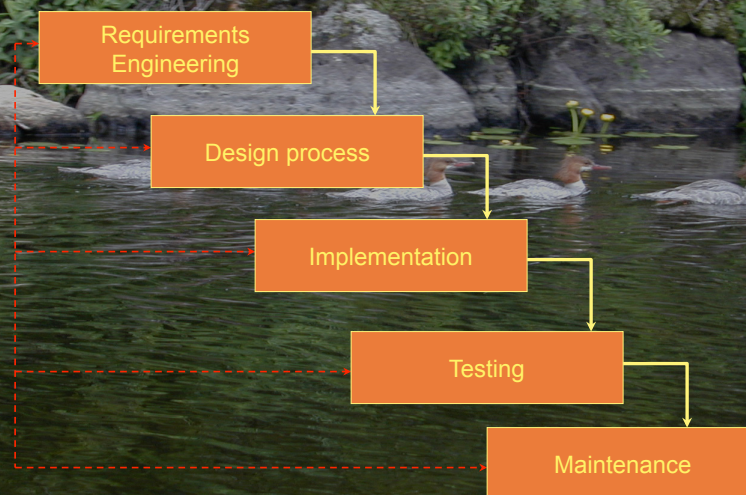
◇ Prototyping models

- Rapid prototyping
- Incremental prototyping
- Evolutionary prototyping
- Spiral model

# Waterfall Model

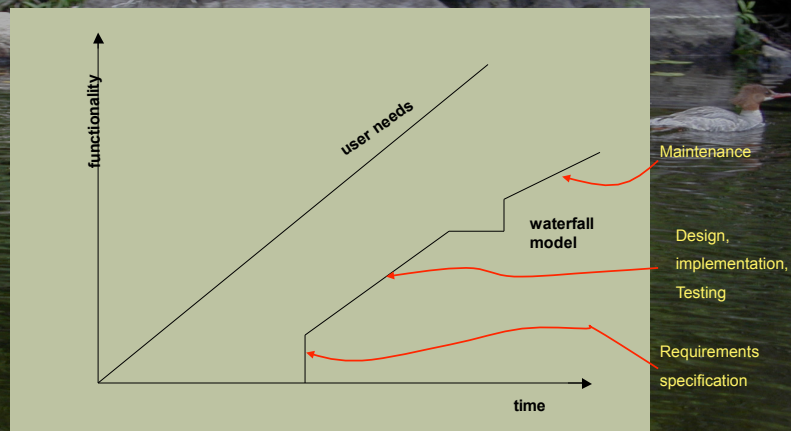
- ◇ Like liquid flows down stair steps ...
- ◇ the liquid flows from one step to the next.
- ◇ Nothing flows uphill without assistance.
- ◇ Work flows through a series of stages (steps)
- ◇ Each stage is completed before proceeding to the next.
- ◇ Previous stages are revisited only when errors occur.

## Waterfall Model (continued)





# Graphical Illustration of Waterfall Model



© Alan Davis, 1990

## Waterfall Model (continued)

### ♦ Advantages

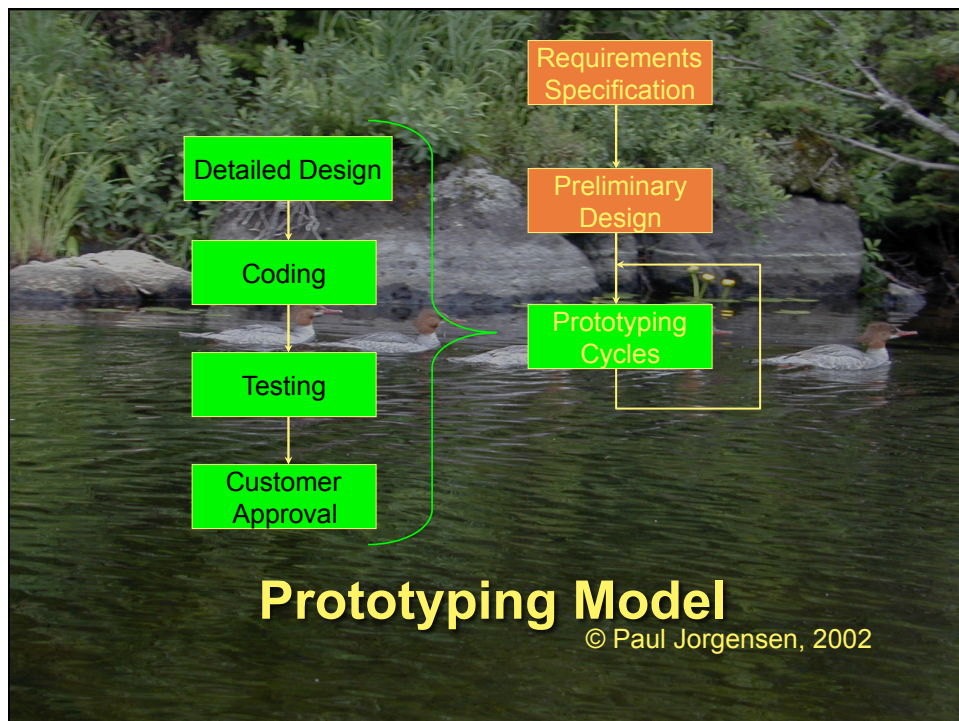
- Linear model; simple and easy to follow
- Feedback from each stage to earlier stages supports flexibility

### ♦ Disadvantages

- Not easy to apply when problems are poorly understood.
- "All requirements must be well understood and frozen before development starts" is unrealistic.
- Estimation of resources requirements is difficult.
- Specific properties of the application domain may be ignored.

# Prototyping Models

- ◇ Repeated (iterated) processes
- ◇ For each iteration
  - Add a (possibly small) subset of requirements
  - Analyze the requirements, design, implement, test (by developer)
  - generally customer must approve before moving to the next iteration
- ◇ Repeat until the product is done







## Advantages of prototyping models

- ◇ Important functionalities can be considered separately.
- ◇ Requirements can be “discovered” at the prompting of an iteration.
- ◇ Customers & managers feel a sense of progress.
- ◇ Prototypes are often used to win customer contracts (especially in bidding).



## Challenges of Prototyping

- ◇ Identifying the subset of requirements for the next iteration can be difficult.
- ◇ Establishing consistency is a repetitive work, particularly when a new subset of requirements bears no relationships with the existing ones
  - Moreover, it is a tedious and time consuming work
- ◇ The project deadline is difficult to predict.
  - No visible end to the set of iterations
  - Poses “maintenance” problems





# Four Prototyping Models

**Rapid prototyping**

**Incremental prototyping**

**Evolutionary prototyping**

**the Spiral model**



## Rapid Prototyping Model

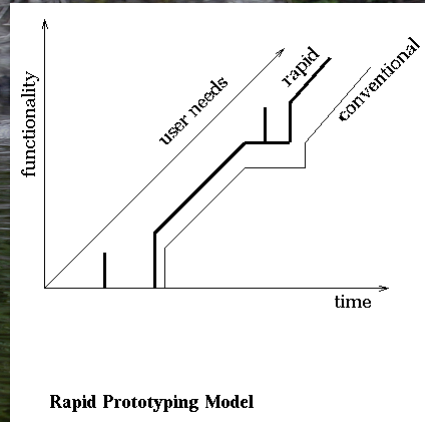
◇ Initial:

- A “quick and dirty” prototype that focuses upon requirements to best impress the customer.
- Development time is a highest priority (“quick”)
- Disciplined design, documentation and testing is often ignored (“dirty”)
- Purpose: to get the approval of the customer

◇ Follow-up:

- After customer approval, much of the prototype is discarded.
- Start the project again from scratch; may use any model
- Use the knowledge and expertise gained in developing the prototype

## Graphical Illustration of Rapid Prototyping Model



## Rapid prototyping model (continued)

◇ Advantages – same as “prototyping models”

◇ Disadvantages

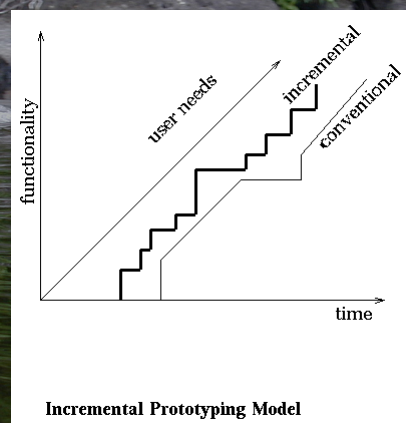
- Time and resources spent on prototype do not contribute to the final product.
- Final product may not resemble the initial prototype.
  - ◇ Developer might have overlooked some features while developing the “quick and dirty” prototype



# Incremental Prototyping Model

- ◇ Initial iteration is similar to rapid prototyping.
- ◇ Following the initial iteration:
  - Developer uses the initial prototype as a base, adding more functionalities.
  - Implementation of the new subset of functionalities should be checked and approved by the customer before proceeding.
  - The developer may use any model to implement the new subset of functionalities -- usually iterative prototyping
  - This is sometimes called the "staircase model".

## Graphical Illustration of Incremental Prototyping Model





## Incremental Prototyping Model

(continued)

### ◇ Advantages

- Similar to “advantages of prototyping models” in general (see previous slides).

### ◇ Disadvantages

- Time consuming
- Can result in too much customer interaction
- No visible end to the finished product

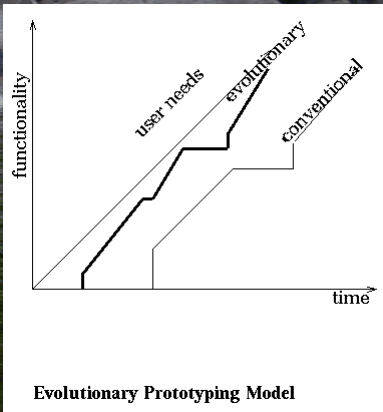
## Evolutionary Prototyping Model

◇ “Initial” and “Follow-up” stages are similar to those of “incremental prototyping”!

### ◇ Major difference:

- Customer interaction is reduced
- Customer consulted only when significant changes are made to the product
- DISADVANTAGE: Who defines “significant change”?

## Graphical Illustration of Evolutionary Prototyping Model



## Spiral Model

- ◇ An evolutionary version of incremental prototyping, proposed by Barry Boehm in 1988.
- ◇ Each iteration of the prototype is represented as a spiral.
- ◇ One full spiral has several cycles
  - Each cycle represents one stage of the development process
  - In each cycle, a set of risks are considered and analyzed at the beginning of the cycle and re-evaluated at the end of the cycle (see the next slide)

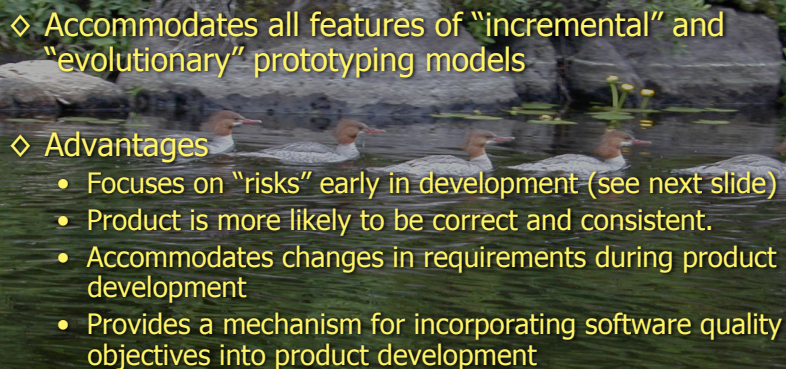


# Spiral Model (continued)



# Spiral model (continued)

- ◇ Accommodates all features of “incremental” and “evolutionary” prototyping models
- ◇ Advantages
  - Focuses on “risks” early in development (see next slide)
  - Product is more likely to be correct and consistent.
  - Accommodates changes in requirements during product development
  - Provides a mechanism for incorporating software quality objectives into product development

- 
- ❖ Accommodates all features of “incremental” and “evolutionary” prototyping models
  - ❖ Advantages
    - Focuses on “risks” early in development (see next slide)
    - Product is more likely to be correct and consistent.
    - Accommodates changes in requirements during product development
    - Provides a mechanism for incorporating software quality objectives into product development



## Spiral model (continued)

### ◇ Risk Management

- Risk identification - a risk is a threat and its consequences
  - ◇ Consider all sorts of "what if ..."
  - ◇ Prioritize/rank risks
- Risk analysis
  - ◇ Identify methods, techniques and schedules to evaluate risks during product development
- Risk control
  - ◇ Evaluate risks during product development and document the results
  - ◇ Plan the next iteration such that the same set of risks do not arise again



## Spiral model (continued)

### ◇ Some development threats listed in literature

- Changes in requirements
- Changes in external resources (e.g., databases)
- Changes in staff/personnel
- Budget overrun
- Resource reallocation
- Equipment failure
- Changes in business policies
- Deviations from expected performance