Umaru Musa Yar'adua University Katsina
Faculty of Natural & Applied Science
Department of Computer Science

CSC4311

SOFTWARE ENGINEERING

LECTURE TWO

LECTURER: DR. AMINU AMINU MUAZU

SOFTWARE PROCESS

Overview

- What is software process?
- Generic process framework
- Examples of process models

Software Process

- Definition:
  - a framework for the tasks that are required to build high-quality software.
- It provide stability, control and organization to an otherwise chaotic activity

Framework Tasks (activities): ⇨

- Communication
- Planning
- Modeling
  - Analysis of requirements
  - Design
- Construction
  - Code generation
  - Testing
- Deployment

What does SW process mean?

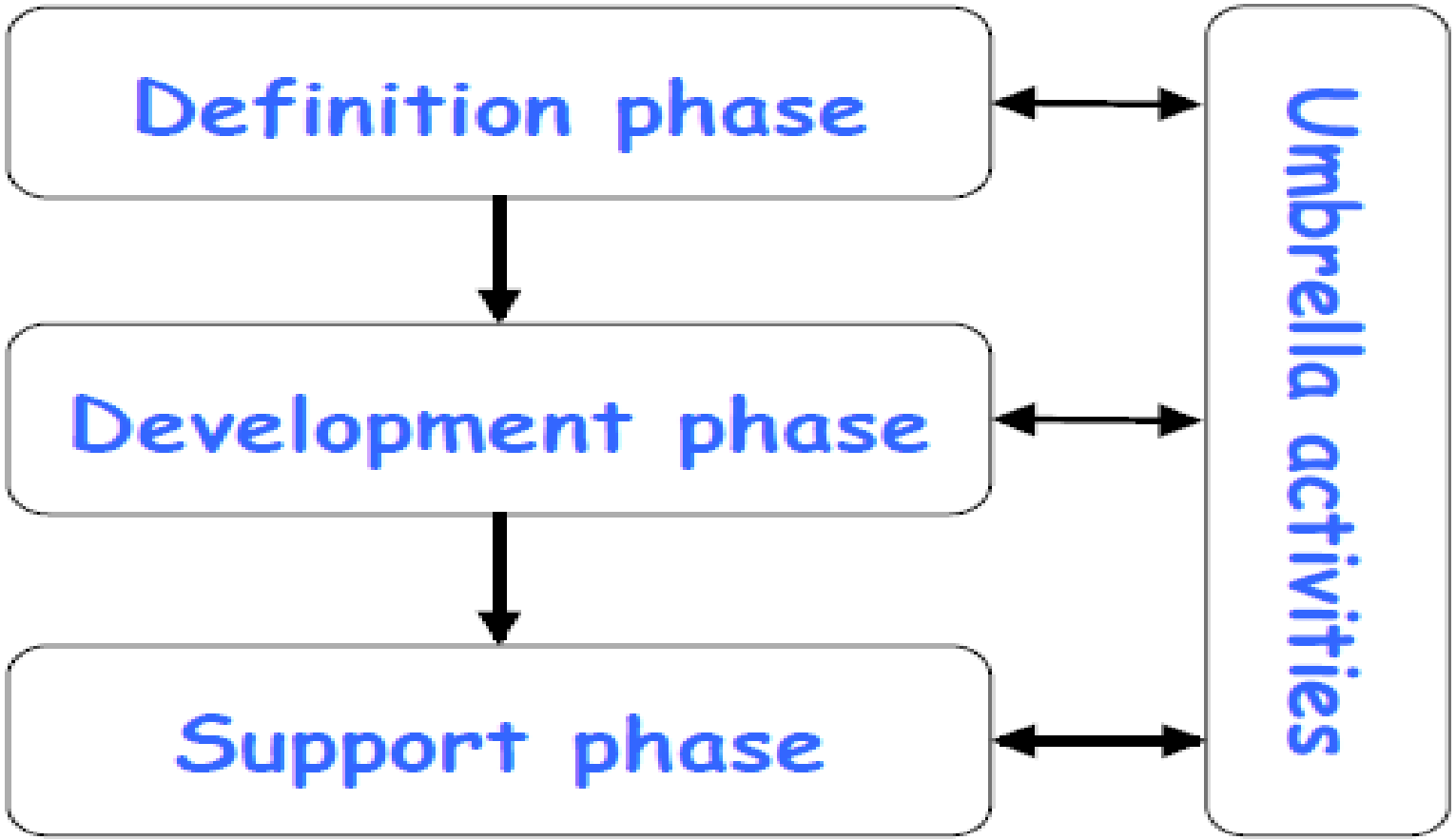
- For a single programmer
  - Design and development
  - Tracking and measuring progress
- For a team of practitioners
  - Organizational planning (time, resources, etc.)
  - Hiring, training, etc.
- For software engineering in general
  - Helps organize SE around ‘best practices’

Elements of SW process

Term	Examples
• People	Software developers, project managers, customers
• Tasks	Analyze requirements
• Planning	Estimate needed resource, time, defects
• Conducting	Track progress and work results

A process defines **who** is doing **what**, **when** and **how** to reach a certain goal.

Generic View of SW Process



Definition Phase

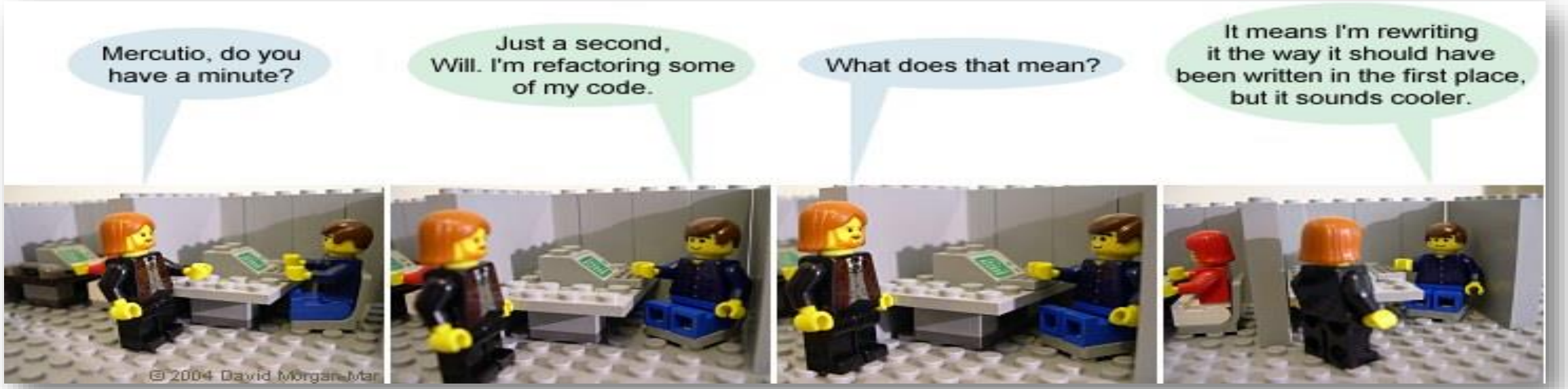
- Tasks related to problem definition
  - What? - requirements, constraints, environment, etc.
  - Step 1: System engineering
    - Ascertain roles of hardware, software, people, databases, operational procedures, etc. in system
  - Step 2: Analysis of the problem
    - Requirement analysis
      - Understanding what the users need and want
  - Step 3: Project planning
    - Resources (e.g., people), cost, schedule

Development Phase

- Tasks related to problem solution
  - How? - architecture, programming, testing, etc.
- Step 1: software design
  - Design models that describe structure, interactions, etc.
- Step 2: code generation/implementation
- Step 3: software testing
  - Goal: uncover as many errors as possible

Support (Maintenance) Phase

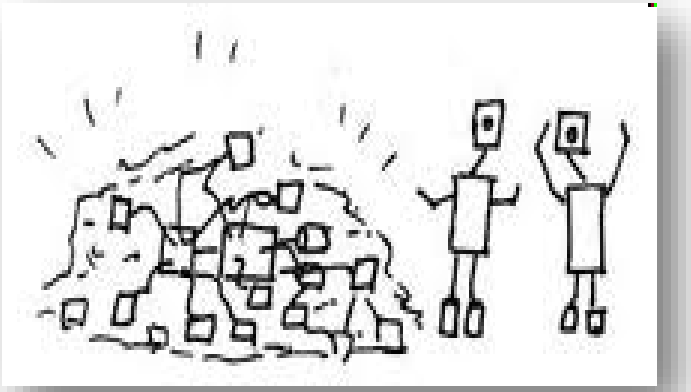
- Tasks related to software evolution
  - Changes? – Definition and development in the context of existing software
- Adaptation to change in the environment
  - New hardware, changes in OS, etc.
- Correction of defects
- Enhancements (new features, etc.)
- Refactoring.



What is Refactoring?

- Is the restructuring (rearranging) code in order to make it easier to maintain and modify.
  - Refactor do not change the code but remove duplications, improve program understanding, improve communications, add simplicity and flexibility.

- When should you refactor?
  - When you add functionality
  - Any time you find that you can improve the design of existing code
  - You detect a “bad smell” (an indication that something is wrong)
  - Examples are:
    - Duplicated code
    - Rename Variable or Method
    - Long method
    - Long parameter list
    - Large class, Etc……..



Some Umbrella Activities

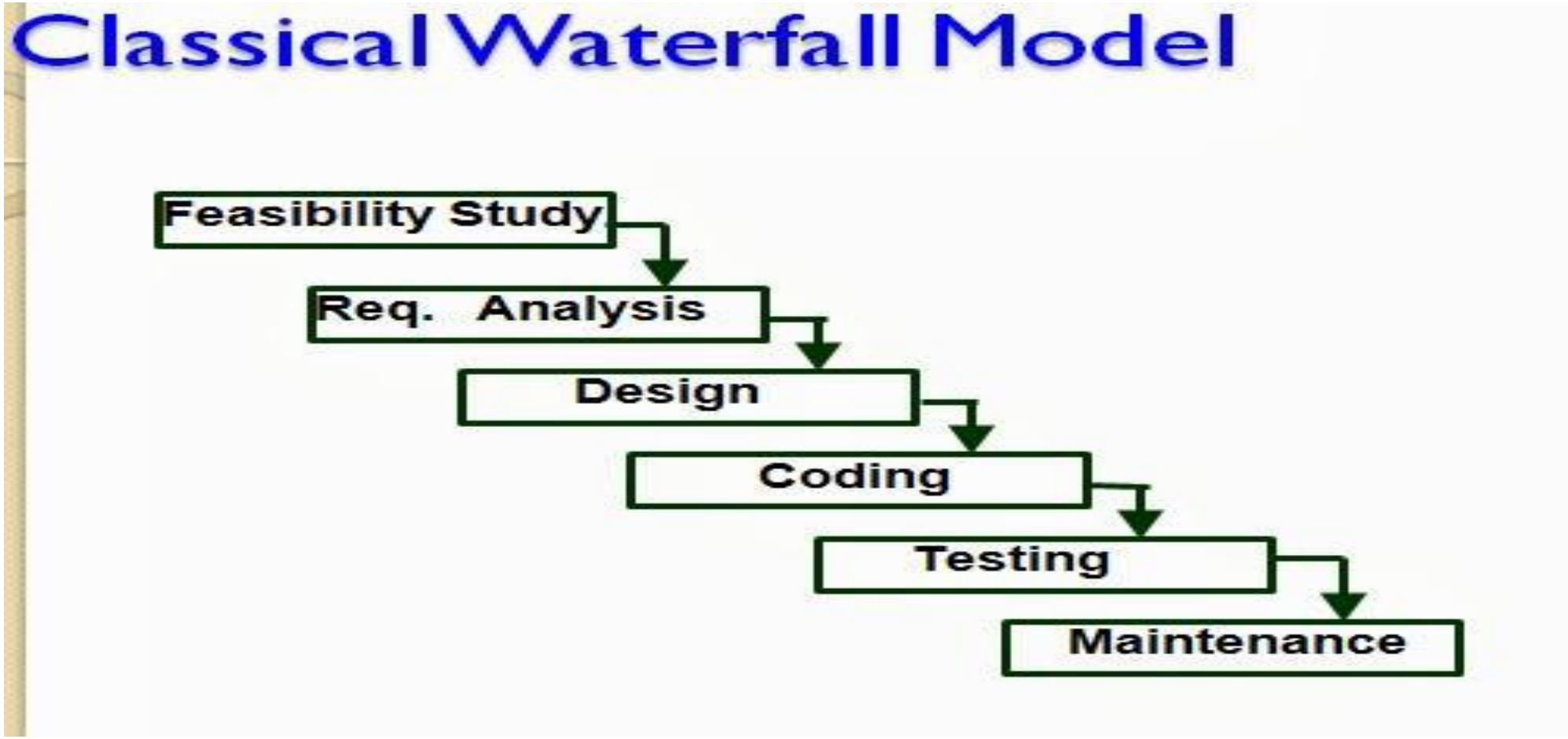
- Project management
  - Tracking and control of people, process, cost, etc.
- Quality assurance (QA)
  - Software testing
- Configuration management
  - Controls the changes in work products.

Examples of Process Models

- Many life cycle models have been proposed so far.
  - Each of them has some advantages as well as some disadvantages.
- A few important and commonly used life cycle models are as follows:
  - Classical Waterfall model
  - Iterative Waterfall model
  - Prototyping model
  - Incremental model
  - Spiral model

Classical Waterfall Model

- The “classic” process model since 1970s
  - Also called “software life cycle”

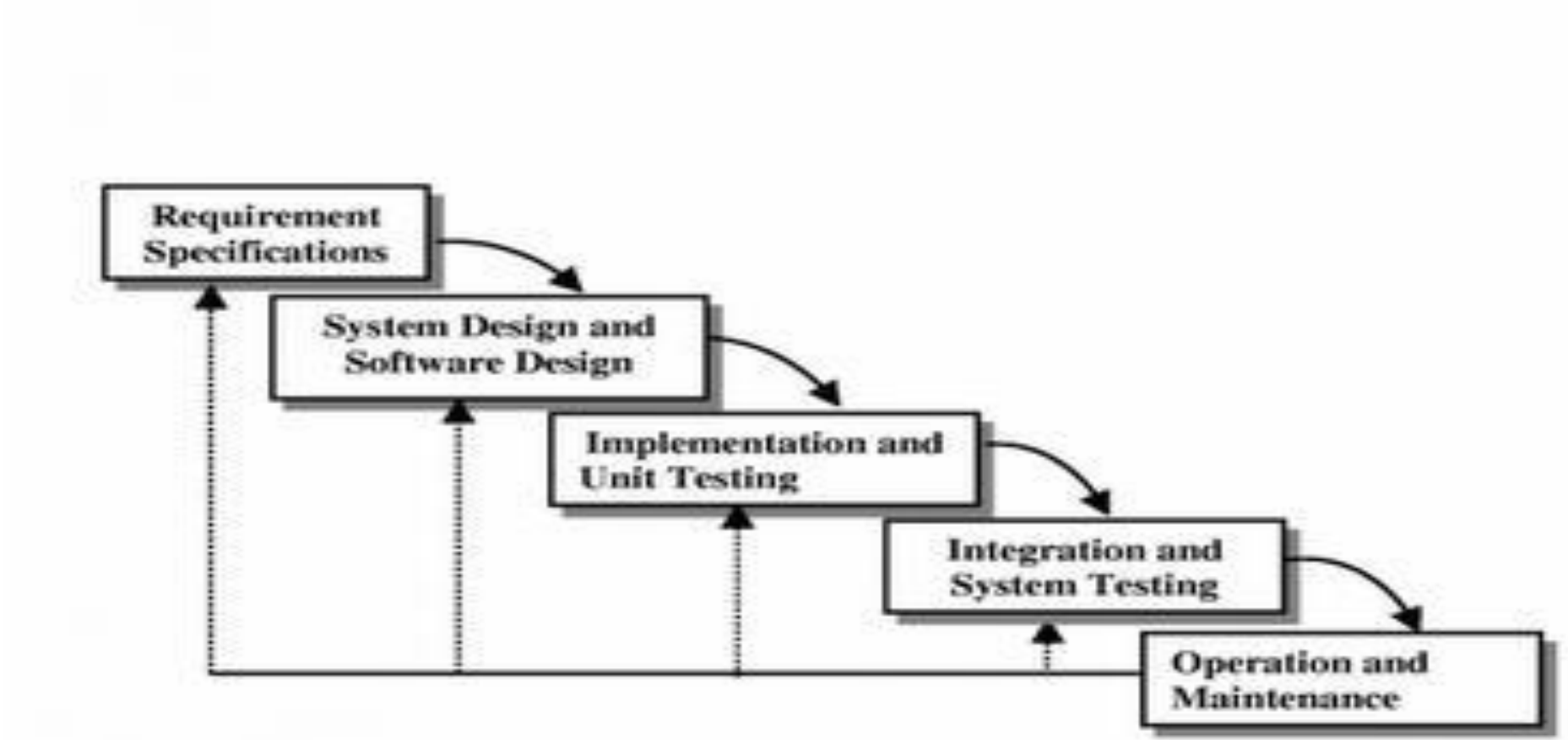


Classical waterfall Phases

- Feasibility study: To determine whether or not to develop the product.
  - Financially and technically
- Requirement Analysis: Define problems
  - requirements, constraints, goals and domain concepts
- Design: Establish solutions into a structure that is suitable for implementation.
  - System architecture with components and their relationship
- Coding: Implement solutions
- Testing and integration: Check solutions
  - Unit testing, system testing
- Maintenance: the longest phase

Iterative Waterfall Model

- The major Shortcomings of classical waterfall model:
  - it assumes that no development error will ever occur
- To overcome the major shortcomings:
  - we come up with the iterative waterfall model.



Key Points of the Model

- The project goes through the phases sequentially
- Possible feedback and iteration across phases
  - e.g., during coding, a design problem is identified and fixed
- Typically, few or no iterations are used
  - E.g., after a certain point of time, the design is “frozen”

Waterfall Model Assumptions

- All requirements are known at the start and stable
- Each phase is completed before the next phase begin
- Testing happens only after implementation
- Minimal customer involvement after the requirement phase

Pros and Cons

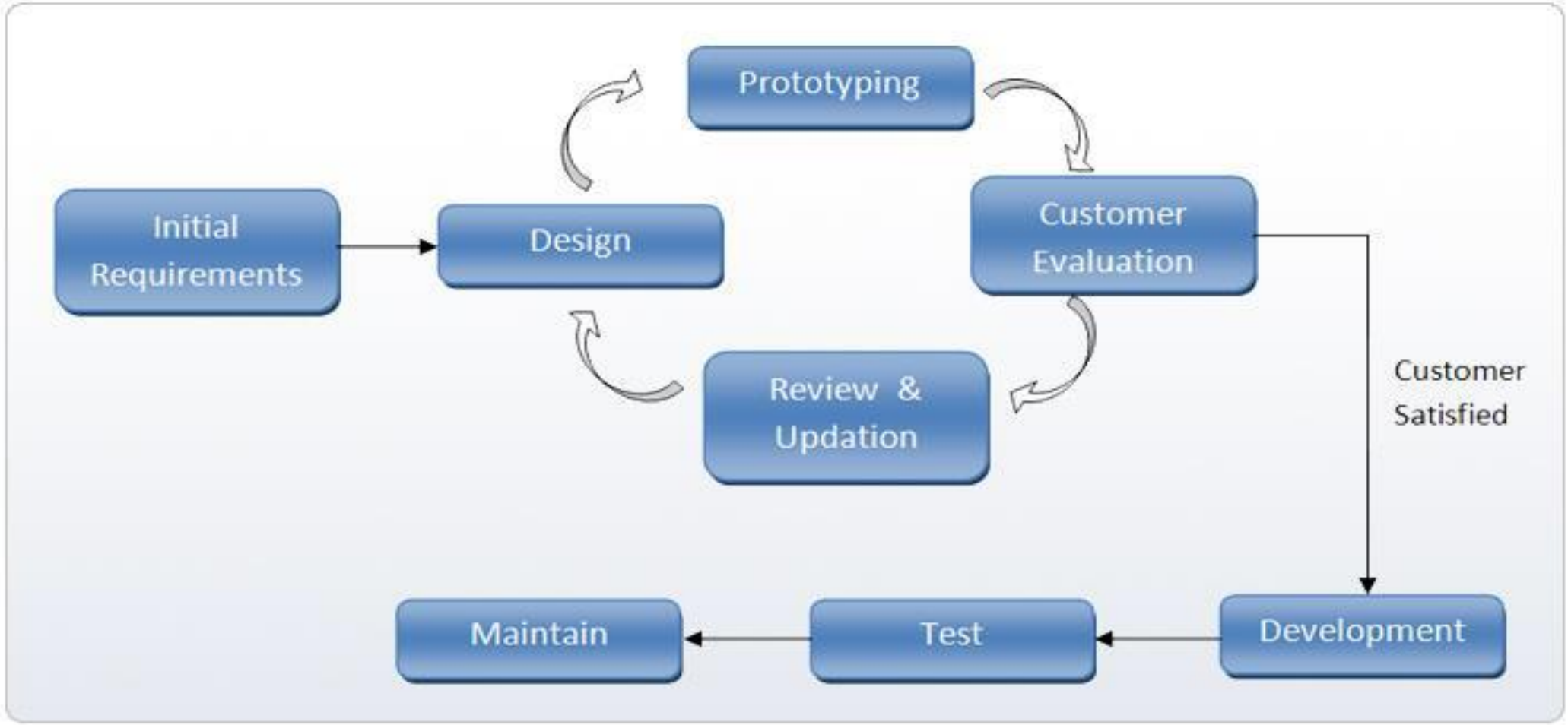
- Pros: widely used, organized, good for projects with well-defined requirements
  - Makes managers happy
- Cons:
  - Applicable only to large and bulky software development projects.
  - The actual process is not so sequential
    - A lot of iterations may happen
  - Expensive and time-consuming

When would you like to use waterfall?

- Work for big clients enforcing formal approach on vendors
- Work on fixed-scope, fixed-price contracts without many rapid changes
- Work in an experienced team

Prototyping Model

- Build a prototype when customers have unclear requirements



Key Points of the Model

- Iterations: Customer evaluation followed by prototype refinement
- The prototype can be paper-based or computer-based
- This is a valuable mechanism for gaining better understanding of the customer's needs:
  - how the screens might look like
  - how the user interface would behave
  - how the system would produce outputs

- Note: the prototype is thrown away!

Pros and Cons

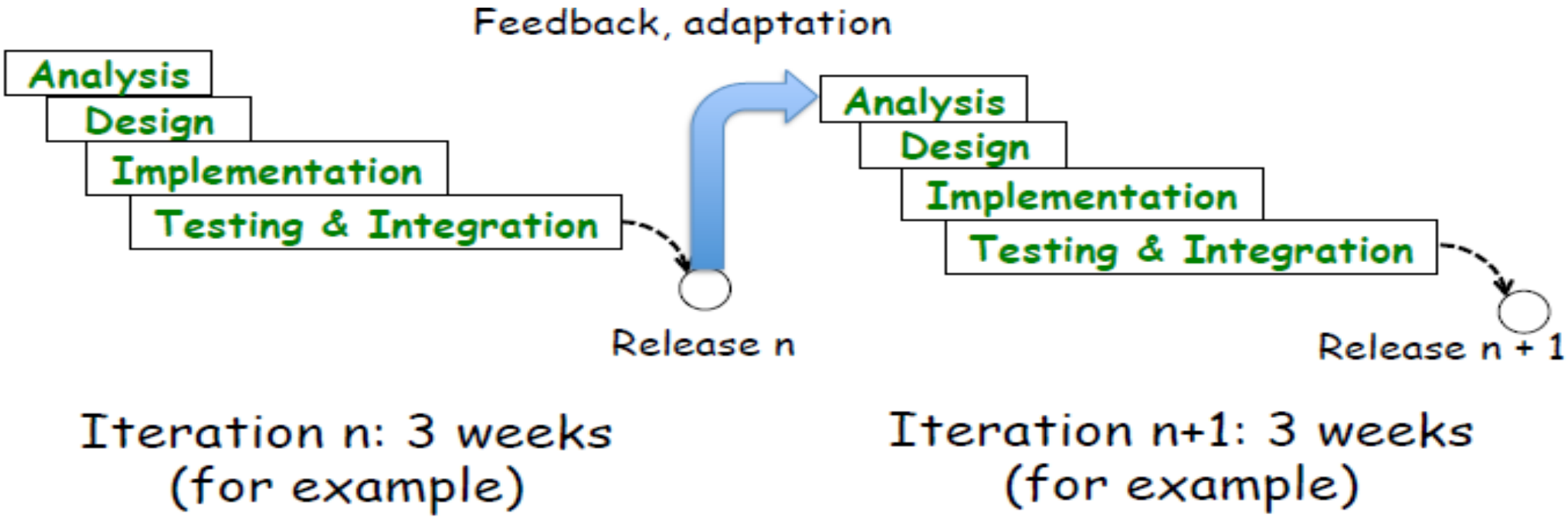
- Pros
  - Facilitate communication about requirements
  - Easy to change or discard
  - Educate future customers
- Cons
  - Iterative nature makes it difficult to plan and schedule
  - Too much investment in the prototype

When would you like to use prototyping?

- When the desired system has a lot of interactions with users

Incremental Model

- A sequential of waterfall models
- also called *successive versions model or Evolutionary model*



Key Points of the Model

- Iterative: many releases/increments
  - First increment: core functionality
  - Successive increments: add/fix functionality
  - Final increment: the complete product

- Require a complete definition of the whole system to break it down and build incrementally

Please what is the core business of Bill Gates, one of the richest man in the world ?

Pros and Cons

- Pros
  - Early delivery of working software
  - User gets a chance to experiment partially developed system
  - Reduce the error because the core modules get tested thoroughly.

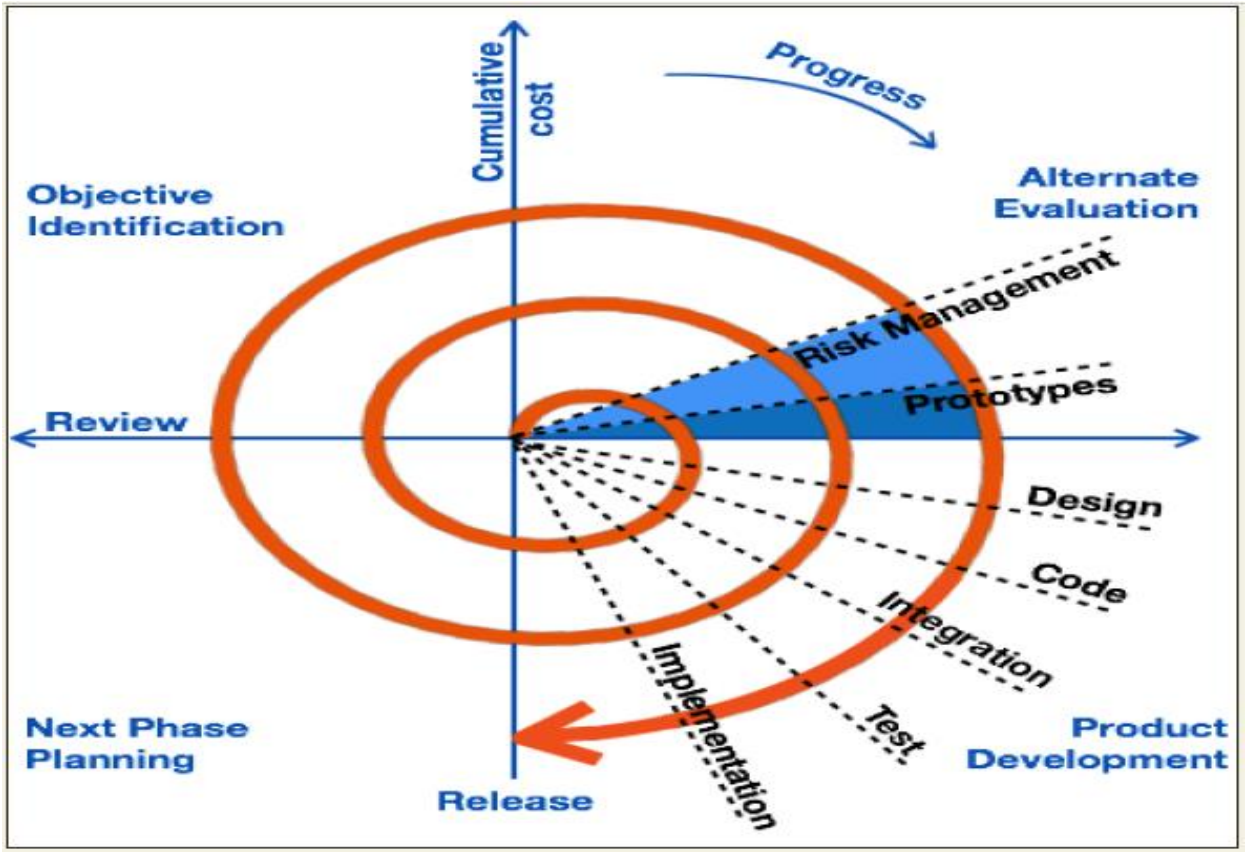
- Cons
  - It is difficult to divide the problem into several versions that would be acceptable to the customer which can be incrementally implemented & delivered.

When to use the model?

- The requirements of the complete system are clear
- Major requirements must be defined while some details can evolve over time
- Need to get a product to the market early

Spiral Model

- A risk-driven evolutionary model that integrates key aspects of development models (waterfall, prototype, etc.)
  - Waterfall model (by including sequential phases
  - Prototyping model (by focusing on building early version
  - Incremental model (by delivering the system in iterations)



Spiral Phases

- Objective setting
  - Define specific objectives, constraints, products, plans
  - Identify risks and alternative strategies
- Risk assessment and reduction
  - Analyze risks and take steps to reduce risks
- Development and validation
  - Pick development methods based on risks
- Review and Planning
  - Review the results achieved so far with the customer and plan the next iteration around the spiral.

What Is Risk?

- Something that can go wrong
  - People, tasks, work products
- Risk management
  - risk identification
  - risk analysis
    - the probability of the risk, the effect of the risk
  - risk planning
    - various strategies
  - risk monitoring

Key Points of the Model

- Introduce risk management into process
- Develop evolutionary releases to
  - Implement more complete versions of software
  - Make adjustment for emergent risks

Pros and Cons

- Pros
  - High amount of risk analysis to avoid/reduce risks
  - Early release of software, with extra functionalities added later
  - Maintain step-wise approach with “go-backs” to earlier stages
- Cons
  - Require risk-assessment expertise for success
  - Expensive

When to use the model?

- Large and mission critical projects
- Medium to high-risk projects
- Significant changes are expected

Comparison of different life-cycle models

Classical waterfall model

- Can be considered as the basic model.
- Cannot be used in practical development projects.
  - Because no mechanism to handle the errors

Iterative waterfall model

- Has overcome the problem in classical waterfall model.
- Is suitable only for well-understood problems.
- It is not suitable for very large projects and for projects that are subject to many risks

Comparison of different life-cycle models

Prototyping model

- Is Suitable for projects for which the user requirements are not well understood.
- Is especially popular for development of the user-interface part of the projects.

Incremental model

- Is suitable for large problems which can be decomposed into a set of modules for incremental development and delivery.

Comparison of different life-cycle models

Spiral model

- It encompasses all other life cycle models.
- Risk handling is inherently built into this model.
- Is suitable for development of technically challenging software products that are prone to several kinds of risks

Reading Assignment

- Read more on:
  - Unified Modeling Language (UML)
  - Unified Process (UP)
  - Agile Software Development