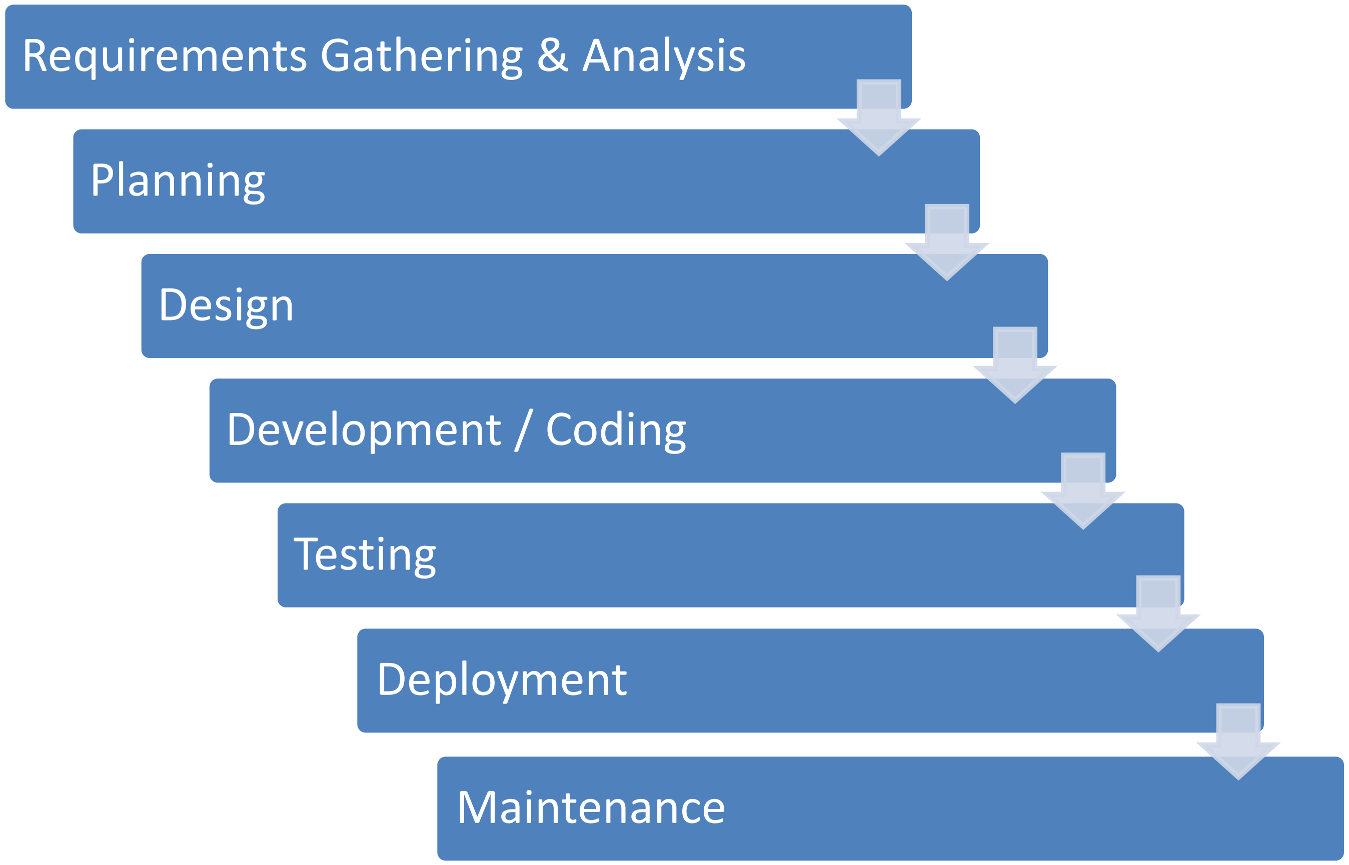
Software Engineering

Vikas Bajpai

Software Development Life-Cycle (SDLC)

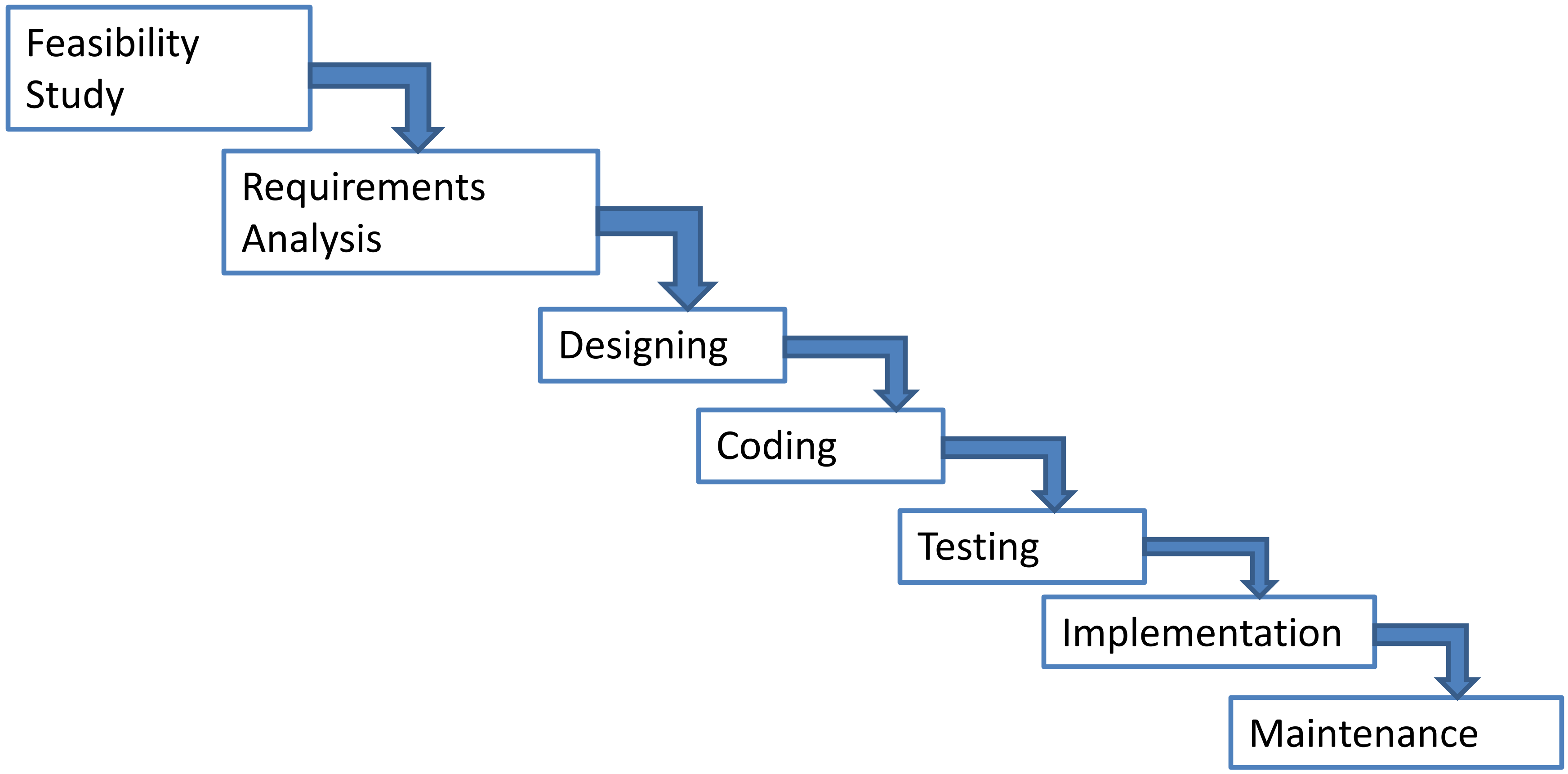
Software Development Life-Cycle (SDLC)



SDLC Models:

1. Waterfall Model
2. Prototyping Model
3. Rapid Application Development (RAD) Model
4. Spiral / Iterative Model
5. V Model
6. Fountain Model

1. Waterfall Model



Advantages of Waterfall Model:

- Easy to understand, easy to use.
- Provides structure to inexperienced staff.
- Milestones are well understood.
- Sets Requirements Stability.
- Good for Management Control (plan, staff, track).
- Works well when Quality is more important than cost or schedule.

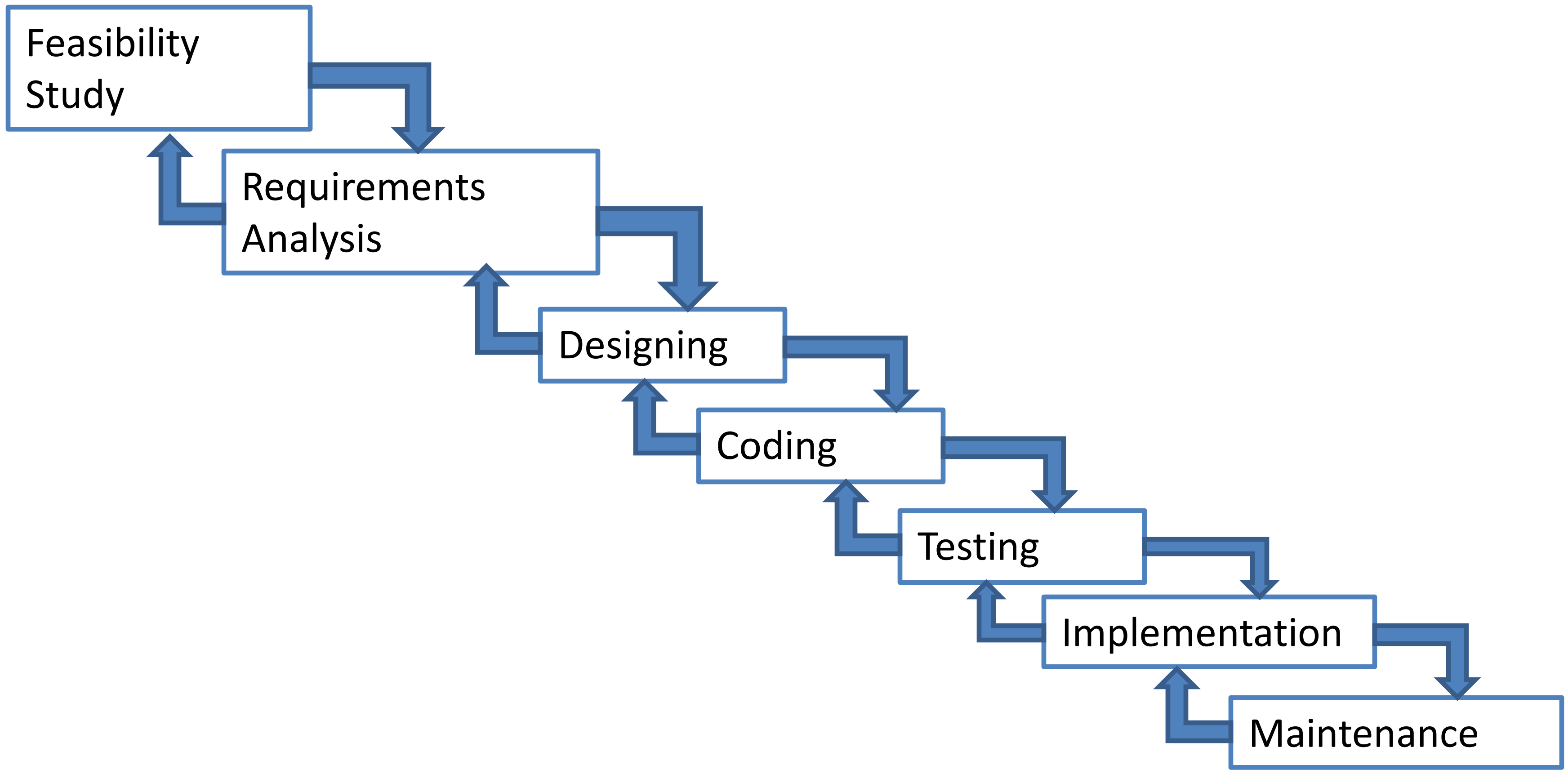
Disadvantages of Waterfall Model:

- No flexibility. Moving back a phase or two can be a costly affair.
- Can give a false impression of progress.
- Does not reflect problem-solving nature of software development – iterations of phases.
- Little opportunity for customer to preview the system.

When to use Waterfall Model

- Requirements are very well known.
- Product definition is stable.
- Technology is understood.
- New version of an existing product.
- Porting an existing product to a new platform.

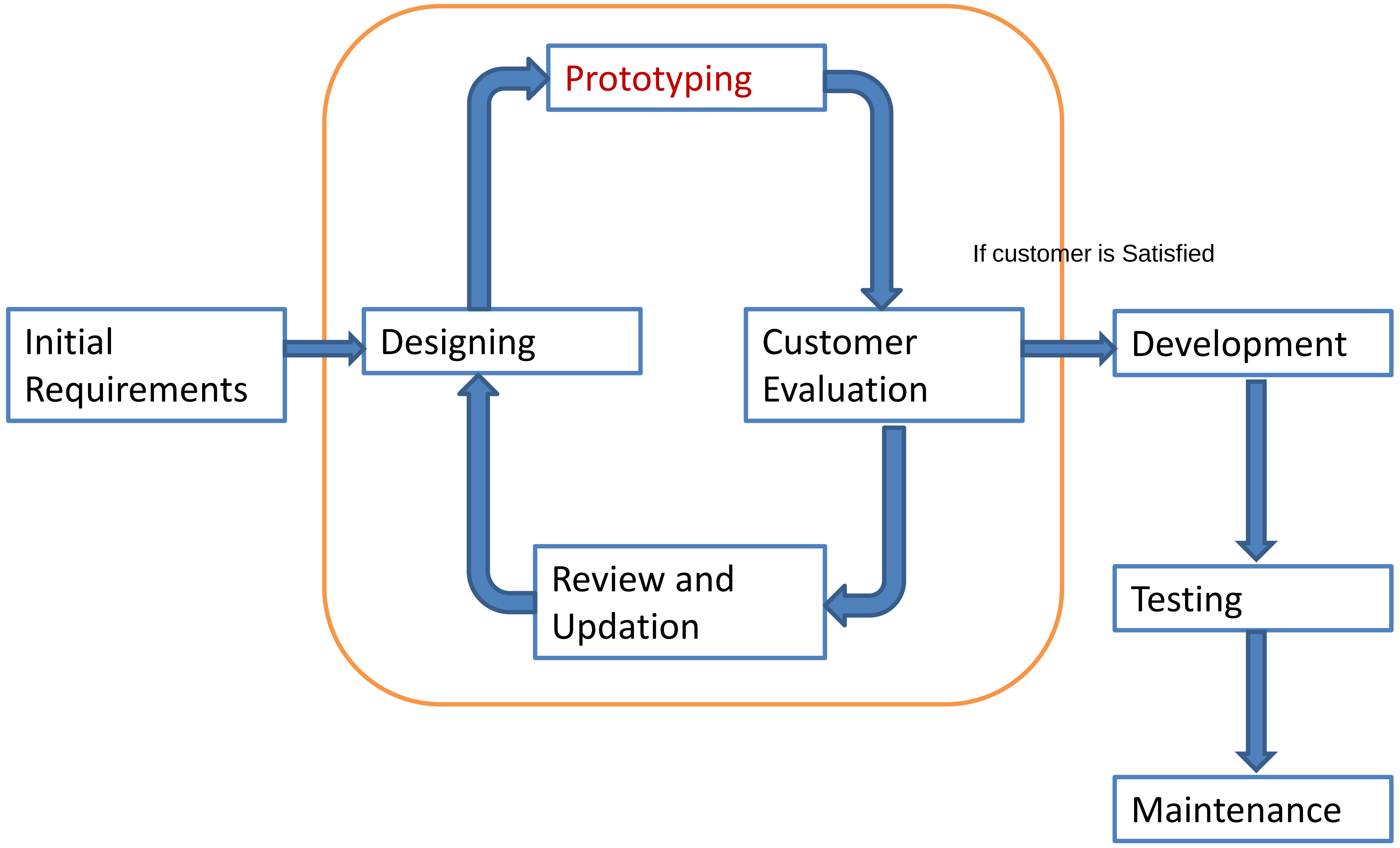
1. a. Iterative Waterfall Model



2. Prototype Model

- Its an information-gathering technique.
- Seeking user reactions, suggestions, innovations, and revision plans / Feedback (if any), from the users.
- Developers build a prototype, which is evaluated by users.
- Refinement of prototype.
- If user is satisfied, then prototype is brought up, for a final product.

2. Prototype Model



Advantages of Prototype model:

- Can bring changes to the system, early in development.
- Addresses users' needs and expectations.
- Ultimately, Customer’s satisfaction is achieved.

Disadvantages of Prototype model:

- Managing the prototyping process is difficult.
- Its rapid and iterative nature.
- Feedback on the prototype.
- Can become a time consuming process.

3. RAD Model

- Much faster development.
- Higher quality results.
- Includes CASE* tools and techniques.
- Includes user-driven prototyping.
- Stringent project delivery.

**Computer Assisted Software Engineering*

- Also known as Computer Aided Software Engineering.
- Aim is to get high-quality, defect-free, and maintainable software products.
- For development and maintenance of Software applications.
- For rapid progress in software development.

What is CASE?

"CASE is the use of computer-based support in the software development process"

--SEI-CMU

CASE….

- CASE tools are used for high productivity.
- Improves quality by Consistency Checking.
- CASE Tools:
  - translators, compilers, assemblers, macro processors, linkers and loaders.
  - program editors, debuggers, code analyzers, and program-pretty printers.
- Examples:
  - RUP (Rational Unified Process)
  - ADDICT (Advance Data Dictionary Tool)

Software Engineering Tools:

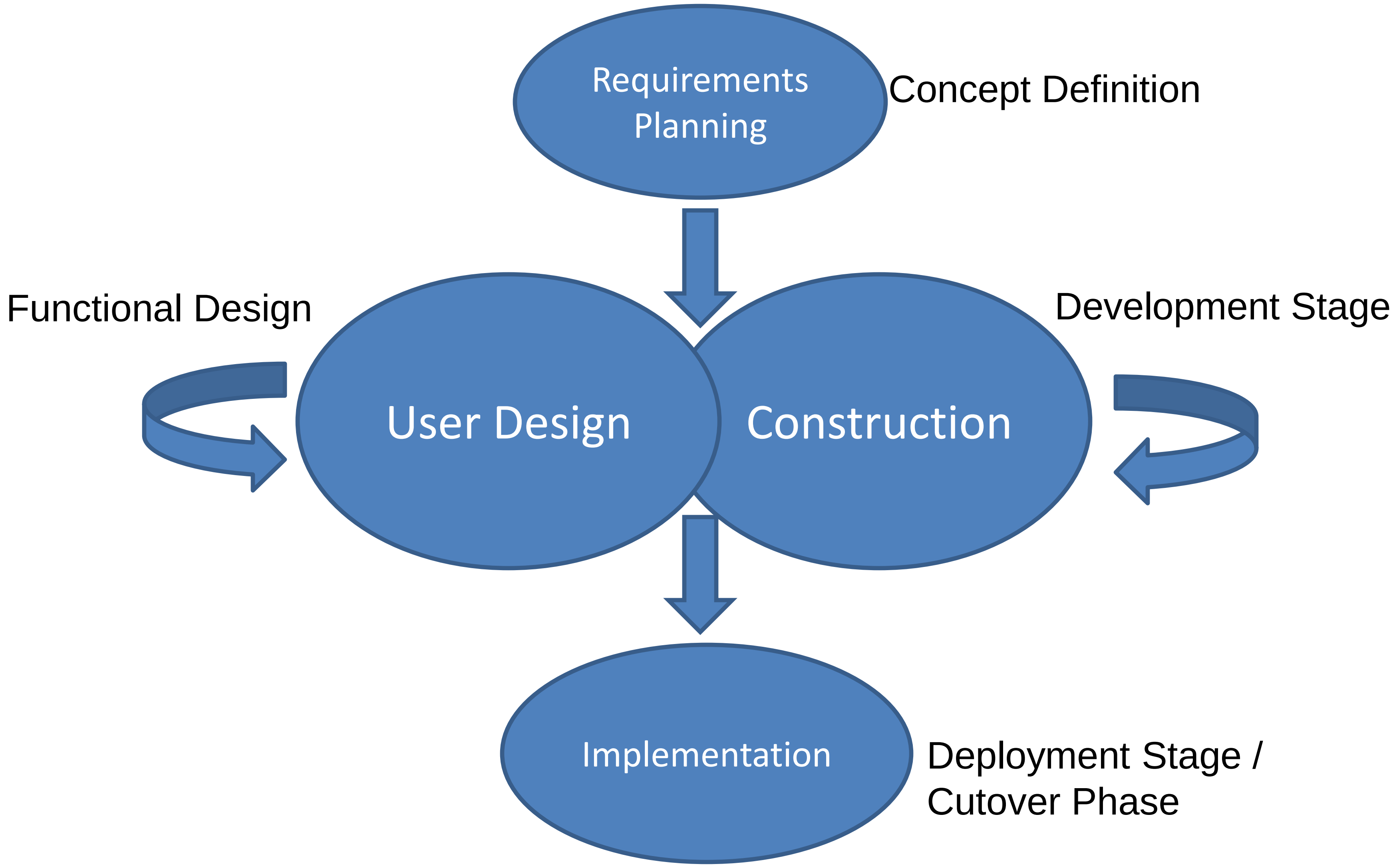
- Programmers Tools
- System Analysis and Design Tools:
  - Point Tools
  - Workbenches
  - Environments

....

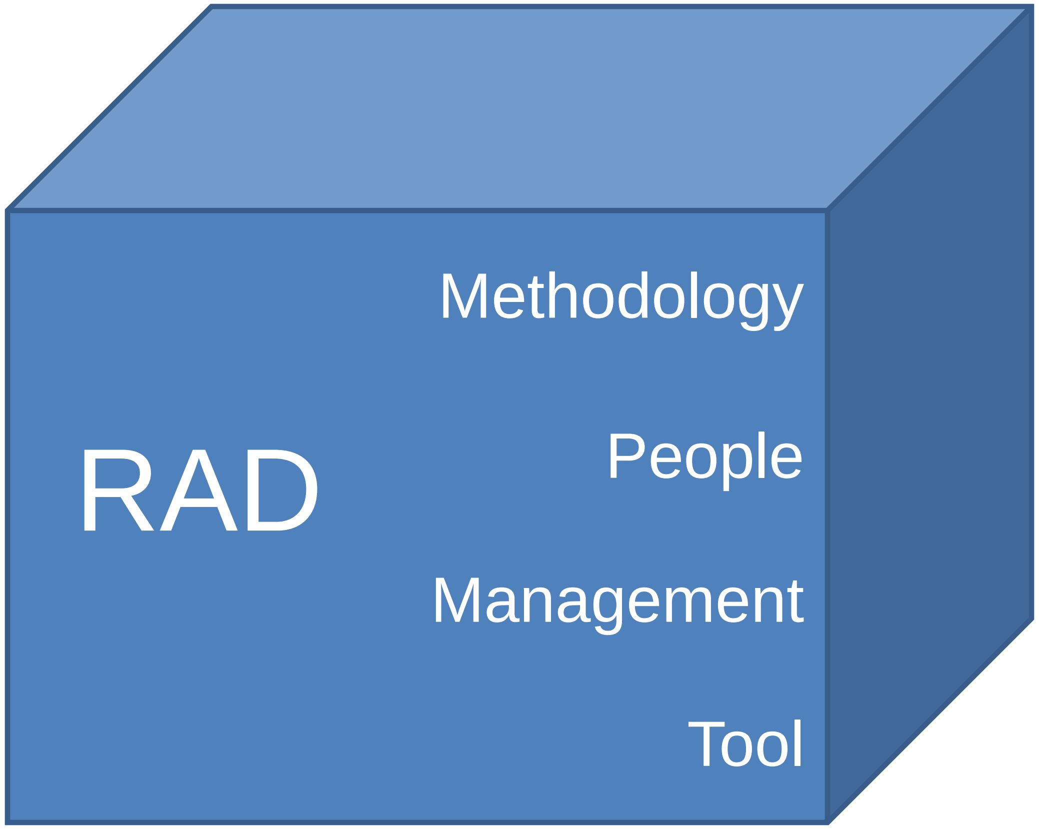
Software Engineering Tools:

- Repositories
- MetaCASE
- Testing tools
- Lifecycle tools

RAD Model


Requirements Planning
Concept Definition
Functional Design
User Design
Construction
Development Stage
Implementation
Deployment Stage / Cutover Phase

Essential Aspects of RAD Model



Essential Aspects of RAD Model

- Methodology
  - Requirements Planning
  - User Design
  - Construction
  - Implementation
- People
  - User Coordinator
  - Requirements Planning Team
  - Training Manager
  - Project Manager
  - Construction (SWAT) Team
  - Workshop Leader
- Management
- Tool

CASE: Verification and Validation

- Tool may be able to automatically detect and resolve inconsistencies in data types or dependencies.
- Design can be automatically generated from requirements.

Advantages of RAD model:

- Less cycles and improved productivity.
- Customer involvement throughout the complete cycle minimizes risk of not achieving customer satisfaction and needs.
- WYSIWYG.

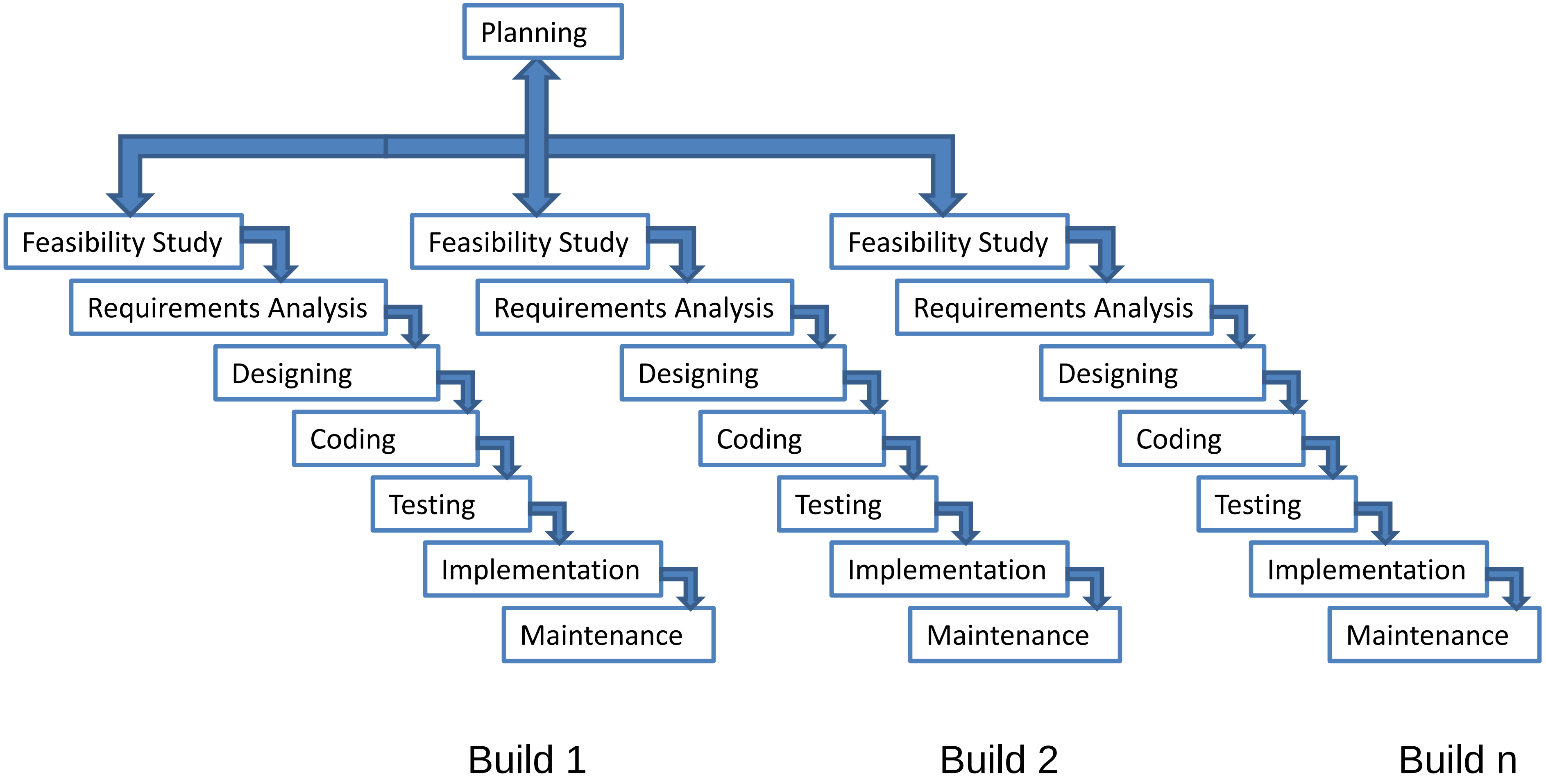
Disadvantages of RAD model:

- Tough to be used with complex systems.
- Need of technically sound Customers.
- Risk of never achieving closure .
- Developers and customers must be committed to rapid-fire activities.

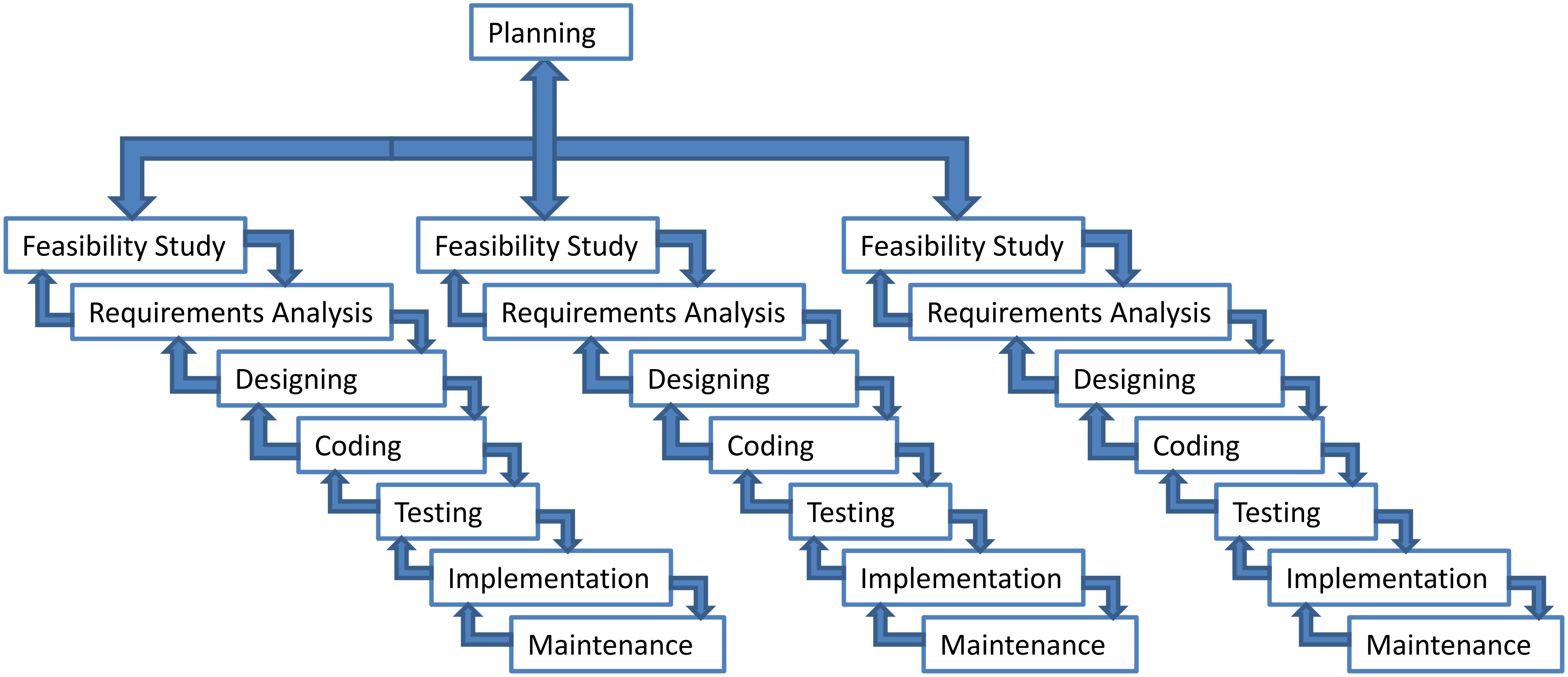
When to use RAD Model:

- Requirements should be well known.
- User’s involvement throughout the life cycle.
- Relatively Smaller Projects.
- High performance not required.
- Low technical risks .
- System can be modularized.

Incremental SDLC Model



Iterative Incremental SDLC Model



Build 1 Build 2 Build n

Advantages of Incremental Model

- Develop risky or major functions first.
- Each release delivers an operational product.
- Customer can respond to each build.
- Approach: “divide and conquer” ie. breakdown of tasks.
- Initial product delivery is faster.
- Customers get important functionality early.
- Risk of changing requirements is reduced.

Disadvantages of Incremental Model

- Requires good planning and design.
- Requires early definition of a complete and fully functional system to allow for the definition of increments.
- Well-defined module interfaces are required (some will be developed long before others)
- Total cost of the complete system is not on the lower side.

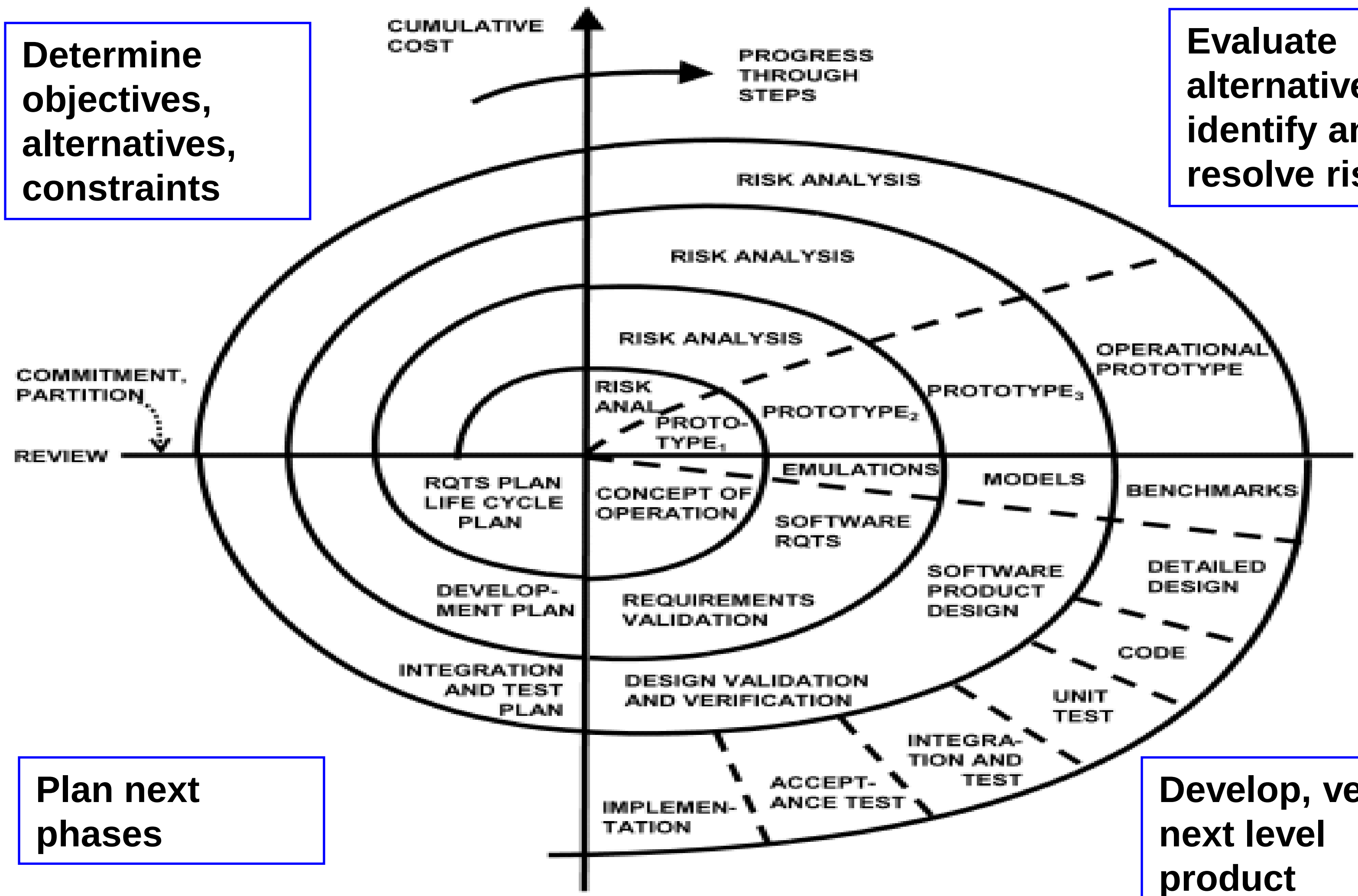
When to use Incremental Model

- Risk, funding, schedule, program complexity, or need for early realization of benefits.
- Most of the requirements are known up-front but are expected to evolve over time.
- A need to get basic functionality to the market early.
- On projects which have lengthy development schedules.
- On a project with new technology.

4. Spiral / Iterative Model

Determine objectives, alternatives, constraints

Evaluate alternatives, identify and resolve risks



Plan next phases

Develop, verify next level product

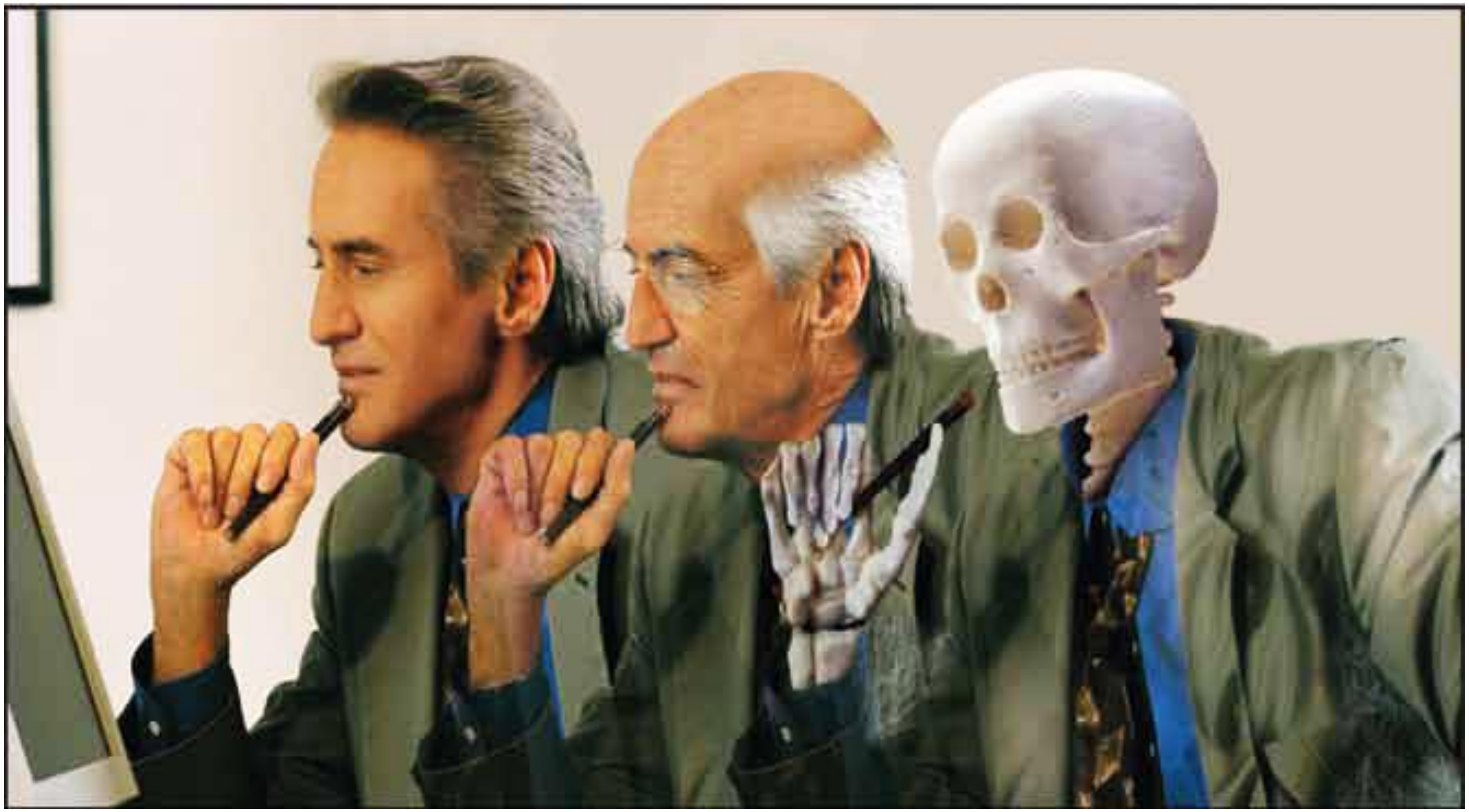
Advantages of Spiral Model:

- Provides early indication of risks.
- Introduces risk management.
- Evolutionary development.
- Release builds for beta testing.
- Users see the system early because of rapid prototyping tools.
- Critical high-risk functions are developed first.
- The design does not have to be perfect .
- Closely tied users to all lifecycle steps.

Disadvantages of Spiral Model:

- Time spent for evaluating risks is too large for small or low-risk projects
- Complex Model.
- Risk assessment expertise is required
- Spiral may continue indefinitely.

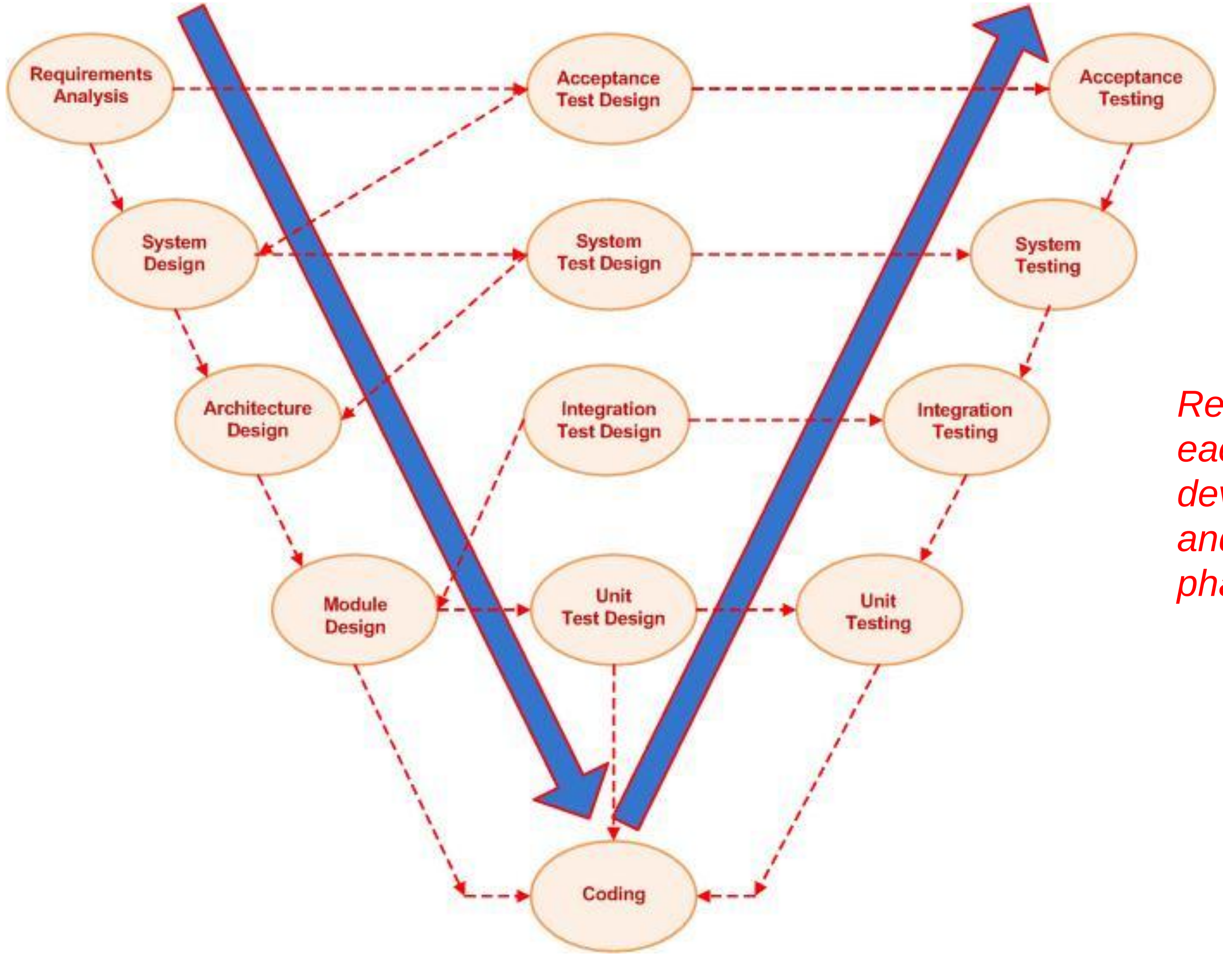
How long can you wait ?



When to use Spiral Model:

- When creation of a prototype is needed.
- When costs and risk evaluation is important.
- For long-term project.
- Users are unsure of their needs.
- New product line.
- Significant changes are expected.

5. V Model



Relationships between each phase of the development life cycle and its associated phase of testing.

Advantages of V Model:

- Planning for verification and validation of the product in early stages of product development.
- Easy to use

Disadvantages of V Model:

- Does not handle iterations.
- No possibility of dynamic changes in requirements.
- No risk analysis activities involved.

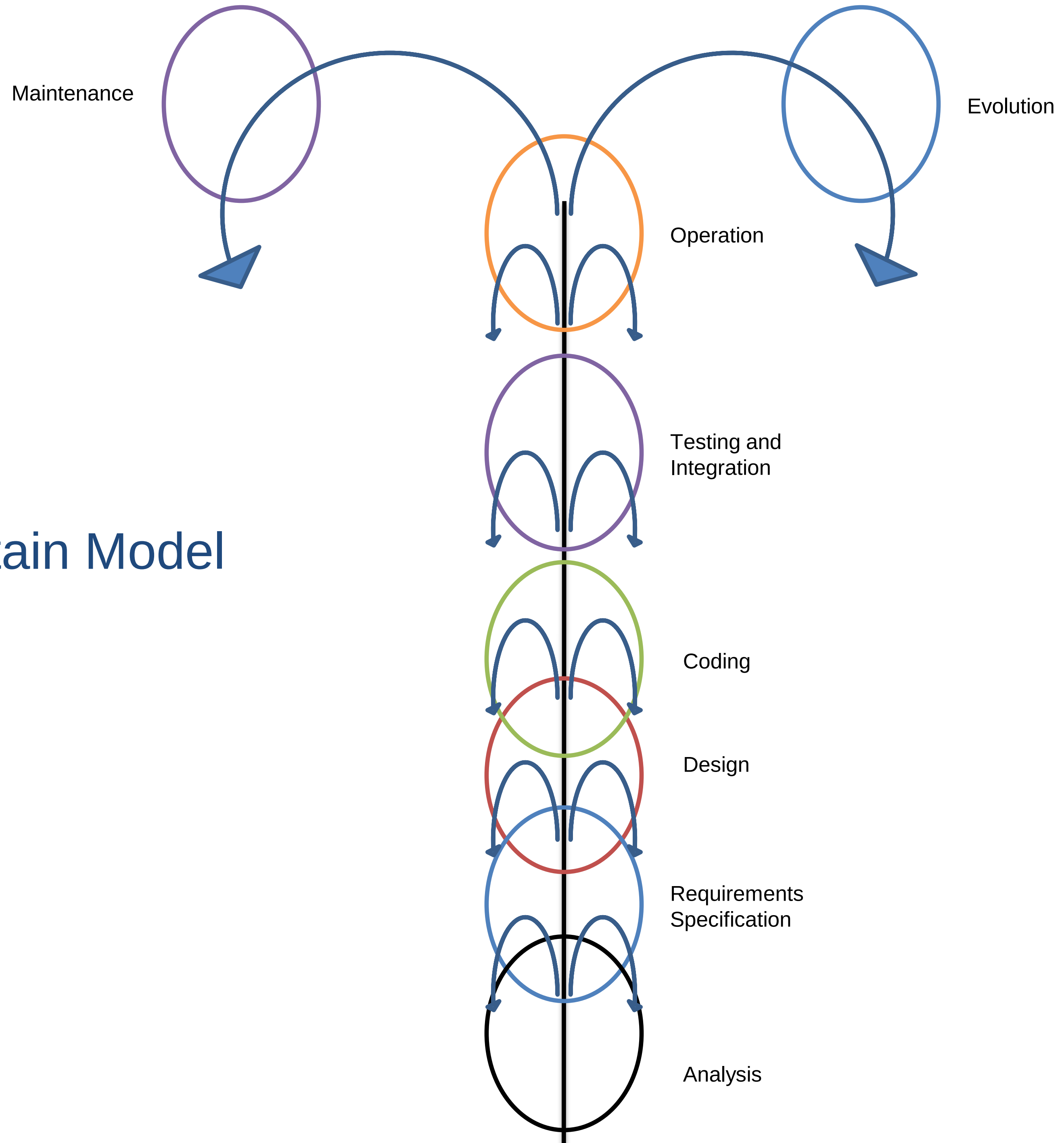
When to use V Model:

- For systems requiring high reliability.
- All requirements are known up-front.
- Solution and technology are known.

6. Fountain Model

- Based on the waterfall model.
- Observes that the sequence always contains cycles.
- Reflects the fact that some phases cannot begin before others.
- A mental image to help visualize what actually happens in many real software development projects.

Fountain Model


Maintenance
Evolution
Operation
Testing and Integration
Coding
Design
Requirements Specification
Analysis