Systems Development Methodologies

- A methodology is a formalized approach to implementing the SDLC.
- System development methodologies:

  I. Traditional (Waterfall, Parallel)
  II. Rapid application development (RAD)
  III. Agile

- **Outsourcing**

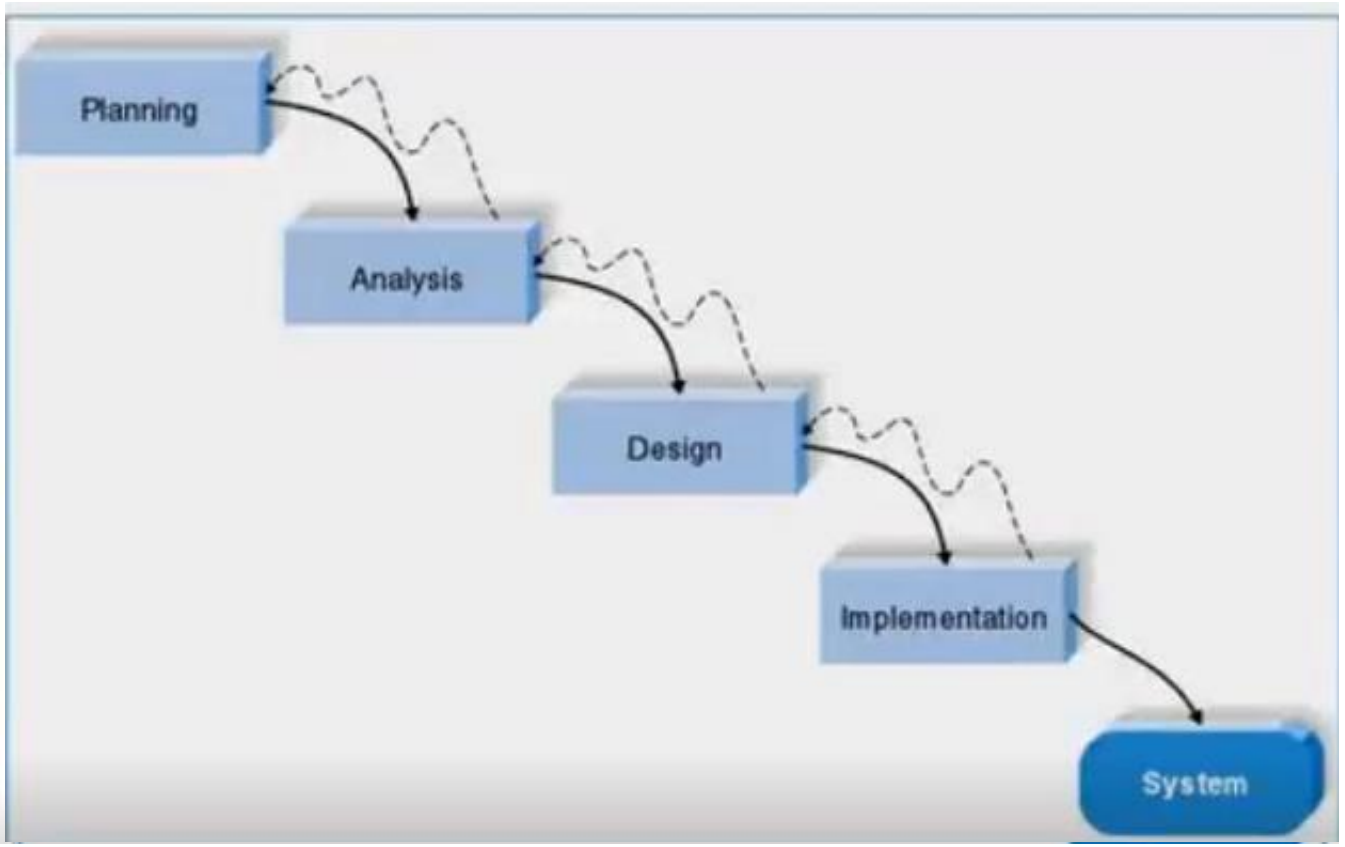
System Development Methodology

Which is the right methodology for your IS project?

- **waterfall** is based on milestone (step by step). The system is produced as valid, and reliable. On the other hand, may one step executes parallel with the previous one. This takes the fundamental process activities of specification, development, validation, and evaluation and represents them as separate process phases such as requirements specification, software design, implementation, testing, and so on.

I. Waterfall Development-based Methodology

With waterfall methodology, the team proceeds sequentially from one phase to the next.



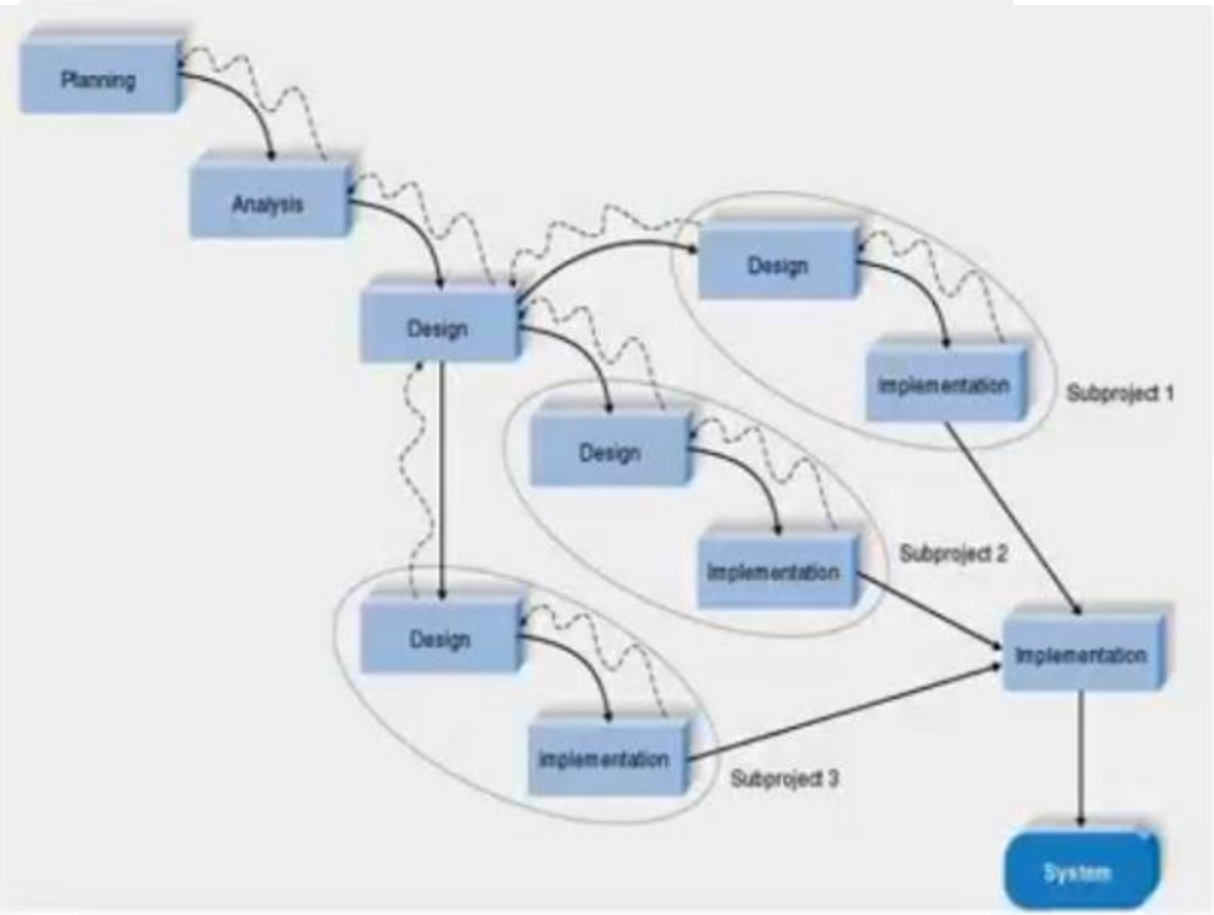
I. Waterfall Development-based Methodology

- **Advantages:**
  - The system requirements are identified long before programming begins.
  - Changes to the requirements are minimized as the project proceeds.
- **Disadvantages:**
  - The design must be completely specified before programming begins (no flexibility).
  - A long time elapses between the completion of the system proposal in analysis phase and system delivery.

I. Parallel Methodology

A general design for the entire system is performed and then the project is divided into a series of distinct subprojects.

This methodology attempts to address the problem of long time between analysis phase and system delivery.



II. Rapid Application Development (RAD)

- RAD is a software development methodology that focuses on building applications in a very short time through iterative prototyping development by automating revisions and changes

- RAD employs tools, techniques and methodologies designed to speed up application development (CASE tools)

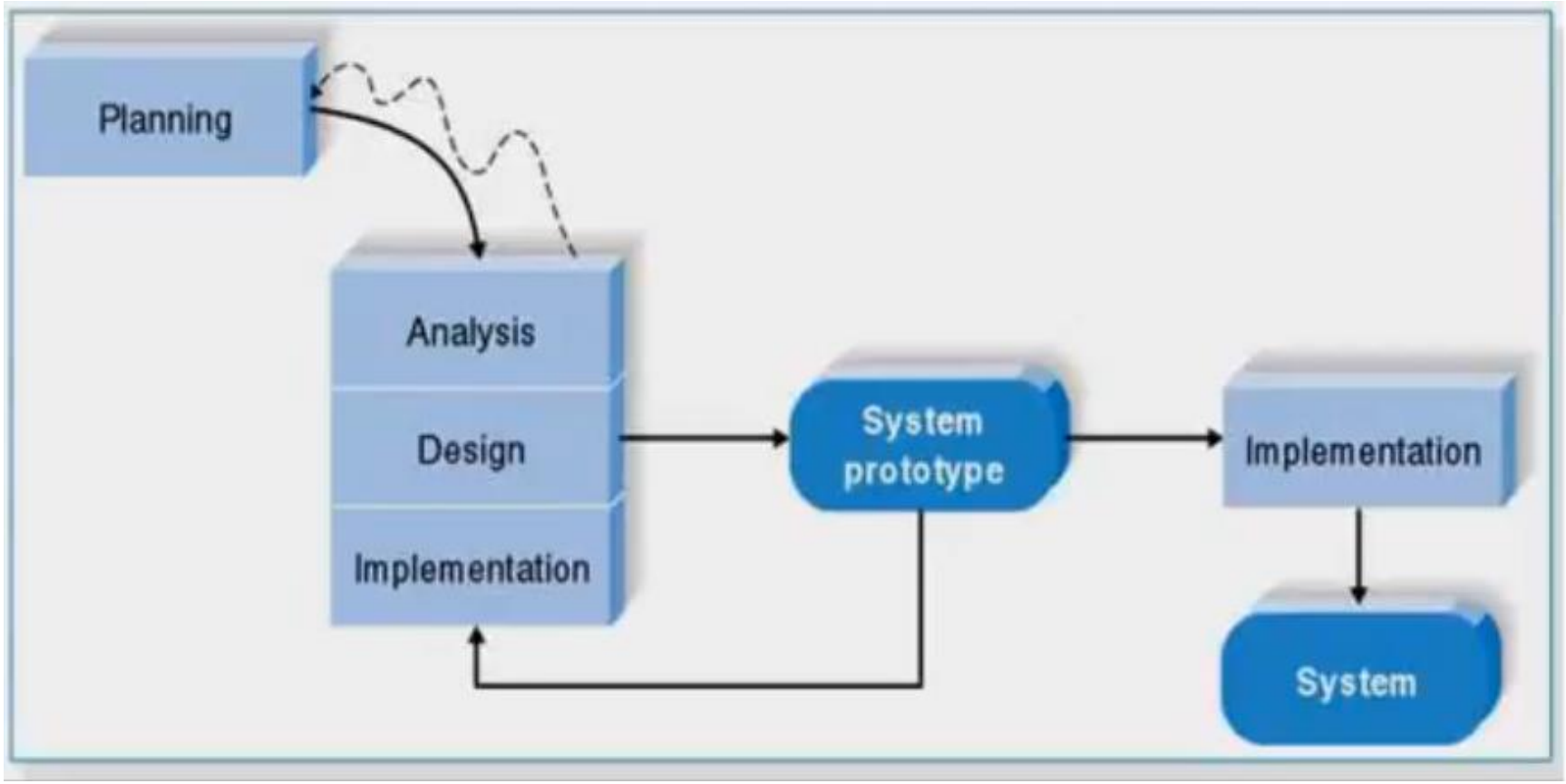
Computer-Aided Software Engineering CASE Tools

– Software tools to automate system development and reduce repetitive work, including
  - Graphics facilities for producing charts and diagrams
  - Screen and report generators, reporting facilities
  - Analysis and checking tools
  - Code and documentation generators

Prototyping

- In prototyping, the analysis, design and implementation phases are performed **concurrently**.
- All three phases are performed **repeatedly** in a cycle until the system is completed.
- A **prototype** is a smaller version of the system with a minimal amount of features
- **Advantage:** Provides a system for the users to interact with, even if it is not initially ready for use.
- **Disadvantage:** Increased development times and often the prototype undergoes significant changes that it cannot be built upon.

Prototyping



Prototyping

- **Operational prototype:**

  - Functioning prototype

  - Accesses real data files, edits input data, makes necessary computations and produces real output

- **Nonoperational prototype:**

A mock-up, or model that includes output and input specifications and formats (UI)

Agile Development

Agile :- actor or customer has a vital role in it to notice the development cycle and make changes.

- is based on the idea of developing an initial implementation, exposing this to user comment and evolving it through several versions until an adequate system has been developed .

- Specification, development, and validation activities are interleaved rather than separate, with rapid feedback across activities.

- is better than a waterfall approach for most business, e-commerce, and personal systems.

- Incremental development reflects the way that we solve problems.

- By developing the software incrementally, it is cheaper and easier to make changes in the software as it is being developed.

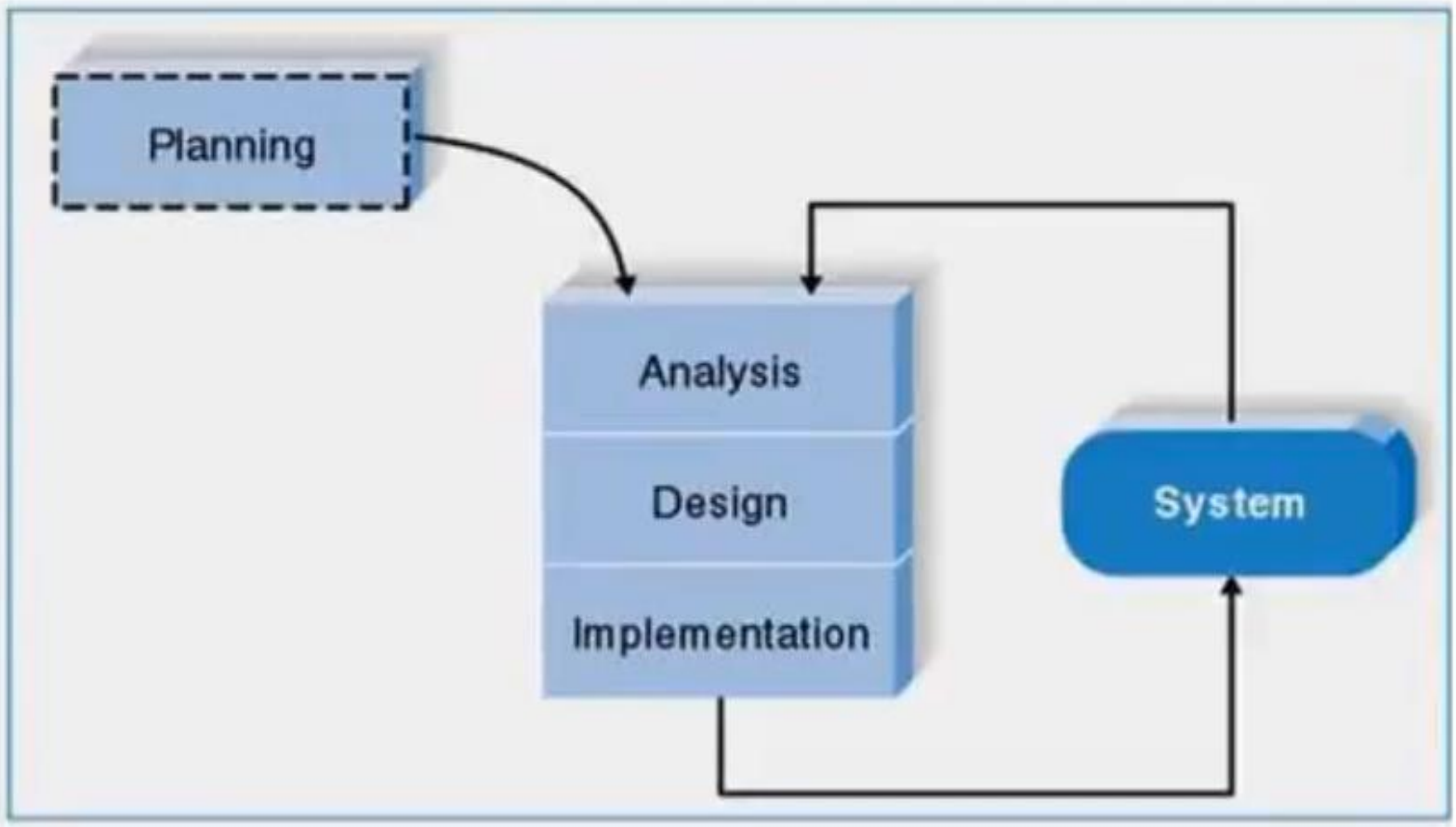
Agile Development

- Focuses on streamlining the SDLC by eliminating much of the modeling and documentation overhead and the time spent on those tasks
- Solves problems of traditional development methodologies
- The most popular agile methodologies include Extreme Programming (XP) and Scrum

Extreme Programming

- Extreme Programming (XP) was founded on four core values:
  - Communication
  - Simplicity
  - Feedback
  - Courage

Extreme Programming



Outsourcing

- Eliminates staffing and personnel problems
- Reduces costs
- Obtains state-of-the-art technology
- Increases technological flexibility

Why IT Development Projects Succeed or Fail?

Success

- User involvement
- Executive management support
- Clear statement of requirements
- Proper planning
- Realistic expectations

Fail

- Lack of user input
- Incomplete requirements and specifications
- Changing requirements and specifications
- Lack of executive support
- Technological incompleteness