



# Process Models

Lecturer: Adel Vahdati



# Generic Process Model - Framework Activities

- Communication
  - Planning
  - Modeling
  - Construction
  - Deployment
- 



# Generic Process Model - Umbrella Activities

- Project tracking and control
- Risk management
- Quality assurance
- Technical reviews
- Measurement
- Configuration management
- Reusability management
- Work product preparation and production



# Generic Process Model - Process Flows

- **Linear**

- Sequential execution of activities from communication to deployment.

- **Iterative**

- Repeats one or more activities before moving to the next.

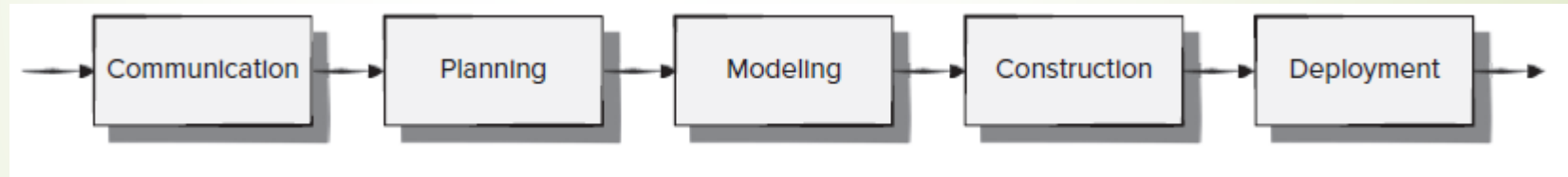
- **Evolutionary**

- Circular execution, leading to progressively complete software versions.

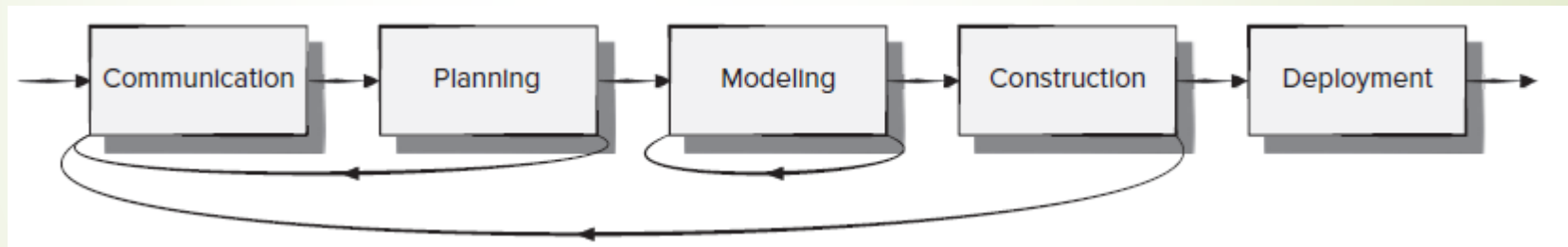
- **Parallel**

- Executes activities in parallel (e.g., modeling and construction).

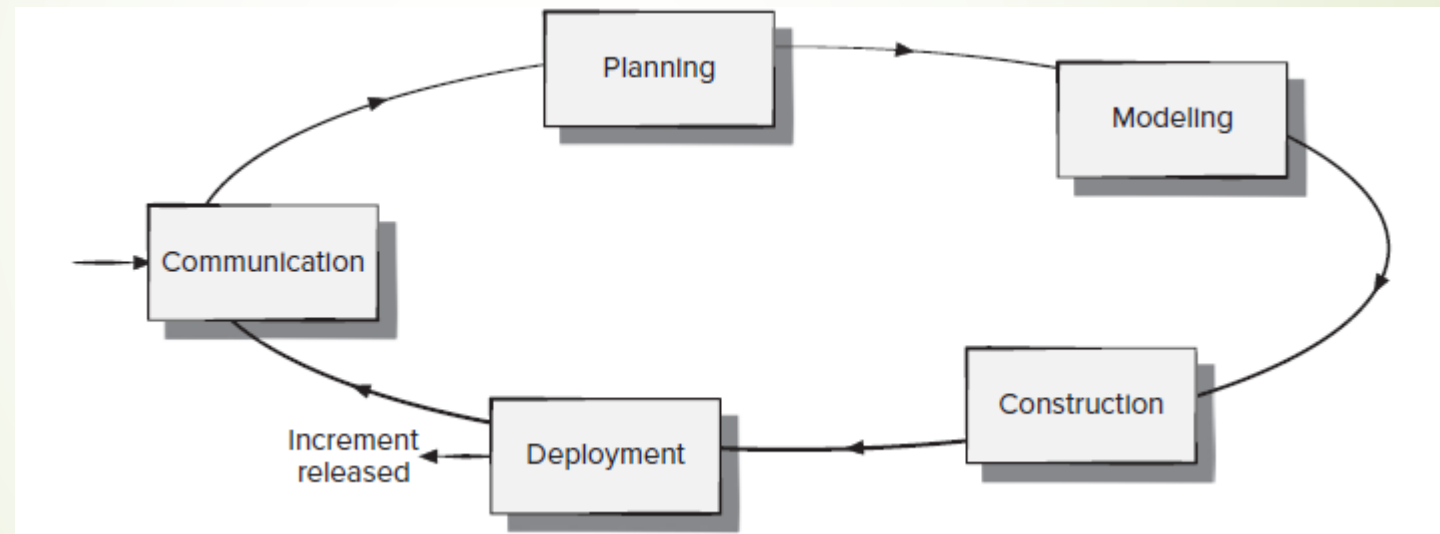
# Linear Process Flow



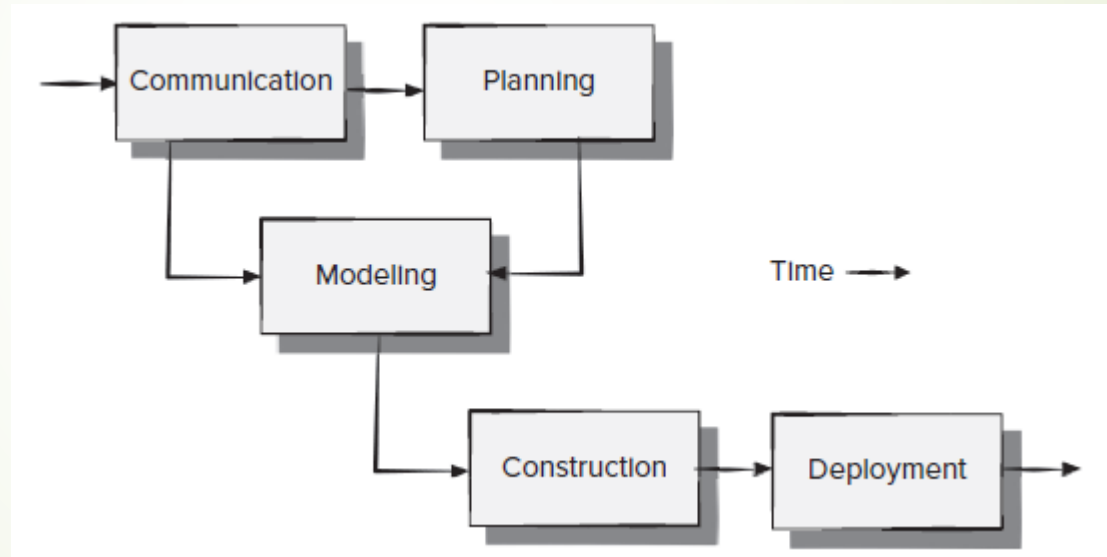
# Iterative Process Flow



# Evolutionary Process Flow

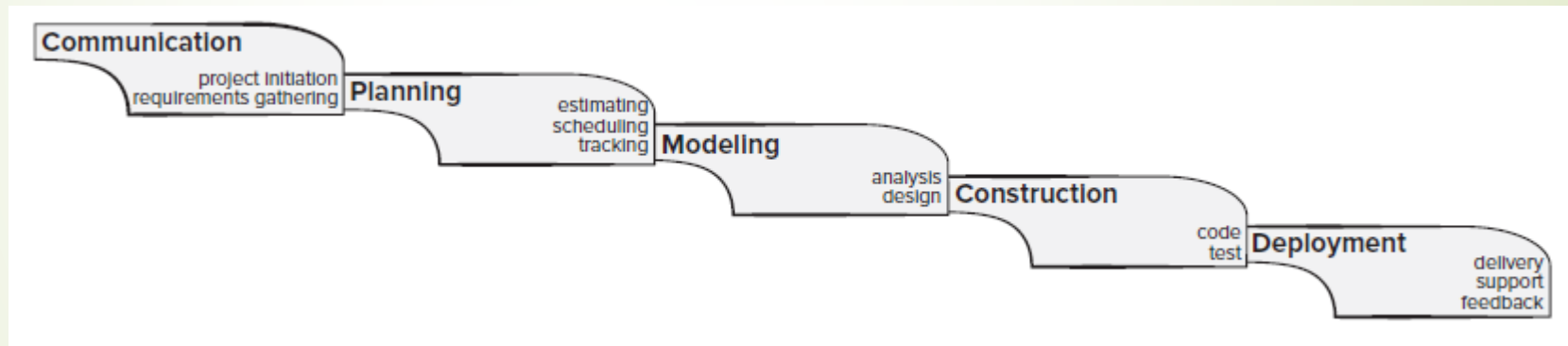


# Parallel Process Flow





# Waterfall Model





# Waterfall Model

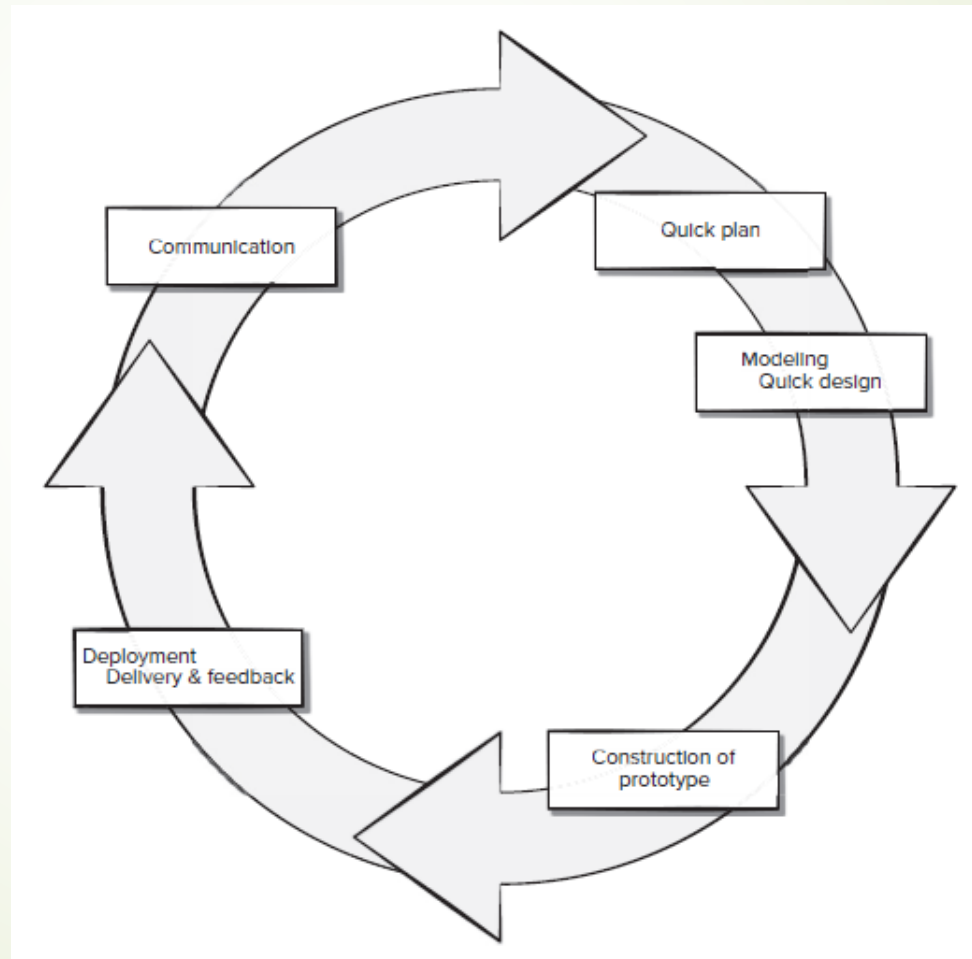
## ➤ Characteristics

- Linear, sequential approach
- Begins with customer requirements, progresses through planning, modeling, construction, deployment, and ongoing support.

## ➤ Challenges

- Rarely follows sequential workflow in real projects.
- Difficult to state all requirements at the beginning.
- Customer must wait for a working version until late in the project.
- Major issues may not be detected until the end.

# Prototyping Process Model





# Prototyping Process Model

## ➤ Characteristics

- Suitable when requirements are unclear or evolving.
- Begins with communication to define objectives and known requirements.
- Quick design and construction of a prototype.
- Prototype evaluated by stakeholders, feedback used to refine requirements.
- Iterative tuning of the prototype to meet stakeholder needs.

## ➤ Challenges

- Stakeholders may mistake the prototype for the final product, unaware of evolving architecture and potential quality issues.
- Developers might make quick, less-than-ideal implementation choices that become part of the final system



# References

- R. S. Pressman and B. R. Maxim. Software Engineering: A Practitioner's Approach. 9th Edition, McGraw-Hill, 2019.
- 