# Waterfall Model

- first, oldest model
- uses the SDLC phases as is (no modification)

Planning ↘
      Requirements ↘
             Software
             design ↘

- Linear, sequential model.
- if previous phase not completed, next phase cannot begin

             software
             implementatn ↘
                Testing ↘
                   Deployment ↘
                      Maintenance &
                      updates
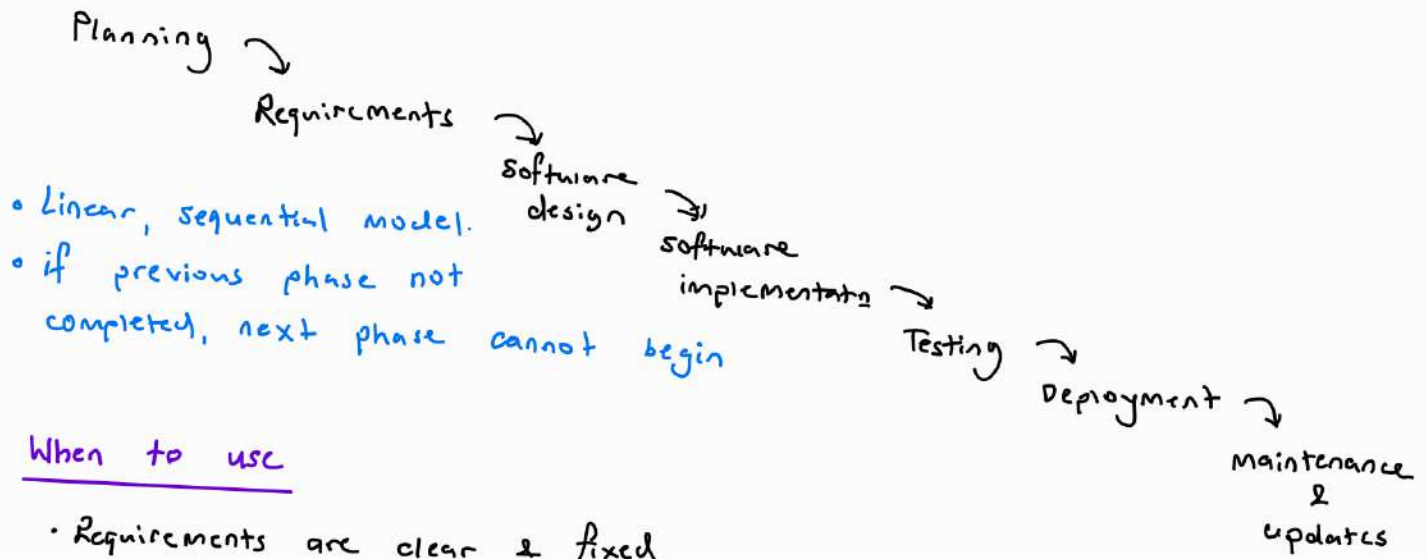
## When to use

- Requirements are clear & fixed
  - ↳ govt. projects
- App not complicated & Big
- Project is short duration
- Enough Resource to execute every stage when needed
- Mission critical projects: Every step must be verified & approved
  - ↳ Low tolerance for bugs

eg:
- E-commerce supply chain app
- payment gateway
- Flight control system
- Stock Trading app

## Pros of waterfall

- Simple & easy to implement (sequential)
- easy to manage
- phases completed sequentially
- scope is fixed (no sudden changes)
- clearly defined stages / Goals
- Good visibility on proj. progress
- Process easily documented

## Cons of waterfall

- No working software at early stages. (customer only see near end of life cycle)
  - ↳ No MVP concept
- Under utilization of resources
  - ↳ eg: coders cannot start until earlier phases complete
- unsuitable if requirements constantly updated
- unsuitable for long term projects
  - ↳ requirements changes
- No scope change possible