Generic Software Process Models

The waterfall model

Separate and distinct phases of specification and development

Evolutionary/Agile development

Specification and development are interleaved

Formal systems development (example - ASML)

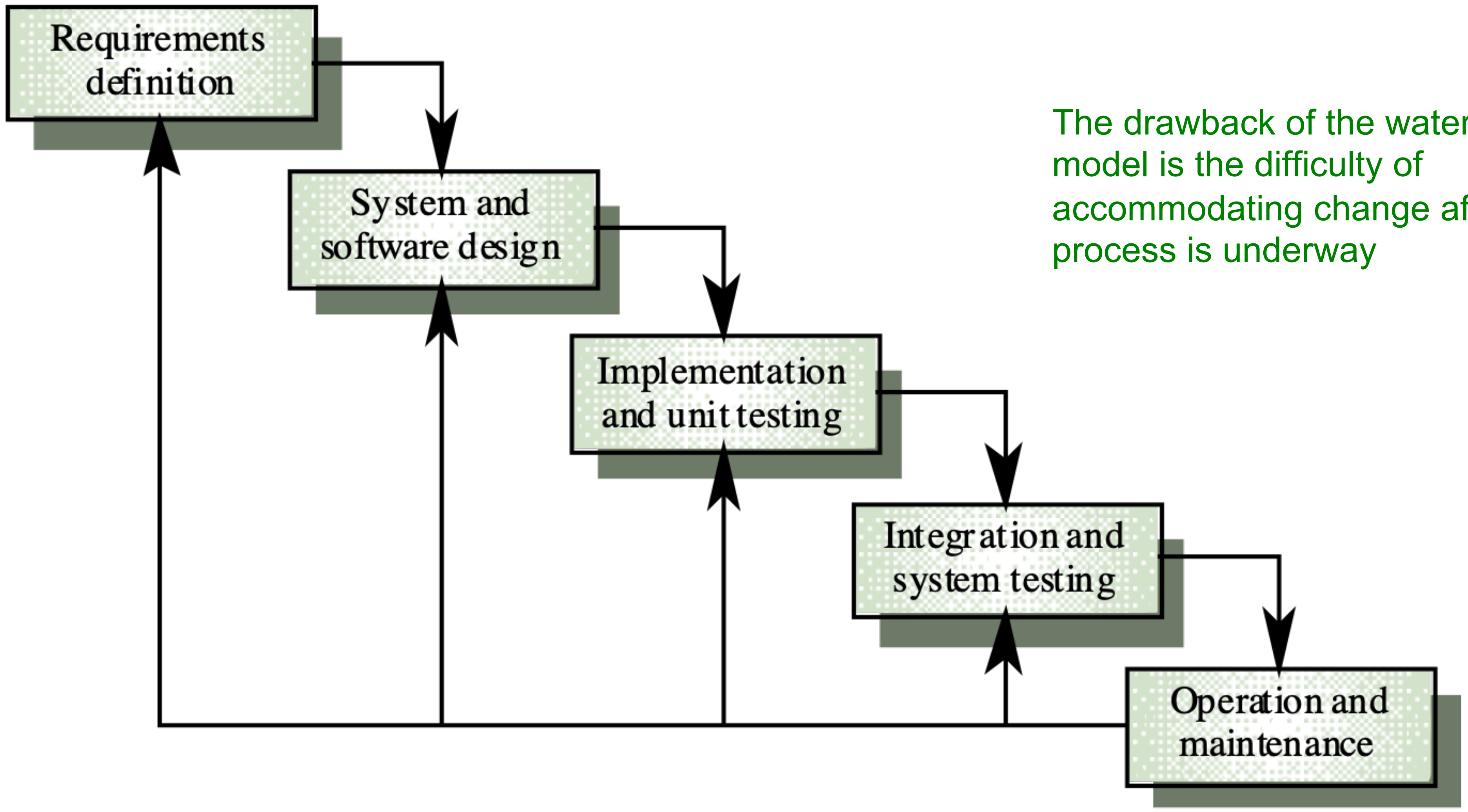
A mathematical system model is formally transformed to an implementation

Reuse-based development

The system is assembled from existing components

1. Waterfall Model

It partitions projects' development into distinct stages


Requirements definition
System and software design
Implementation and unit testing
Integration and system testing
Operation and maintenance

The drawback of the waterfall model is the difficulty of accommodating change after the process is underway

Waterfall model problems

- **Inflexible partitioning** of the project into distinct stages
- This would make it difficult to accommodate changing customer requirements
- **Applicability:** This model is, thus, only appropriate:
  - when the requirements are well-understood at the project start
  - Large and complex systems (too expensive to use for small systems)

Waterfall model describes a process of stepwise refinement

- ➢ Based on **hardware engineering models**
- ➢ Widely used in **military** and **aerospace** industries, where requirements early are well defined and no change in requirements or change is minimal.

Why Not Waterfall

But software is different :

- **No fabrication step**
  - Program code is another design level
  - Hence, no “commit” step – software can always be changed..!
- **No sufficient body of experience for design analysis**
  - Most analysis (testing) is done on program code
  - Hence, problems not detected until late in the process
- **Waterfall model takes a static view of requirements**
  - slow and expensive to changing needs
  - Minimal user involvement after specification is written
- **Unrealistic separation of specification from the design**
- **Cannot easily utilise prototyping, reuse, etc.**