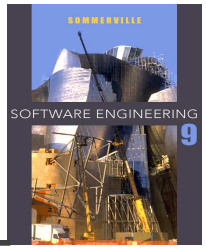
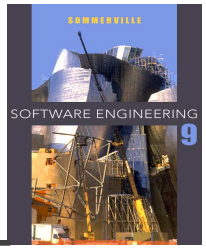


Chapter 2 – Software Processes

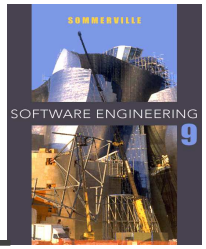
2.3 Coping with Change

Overview:



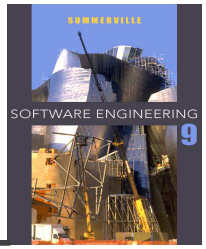
- 2.3 Coping With Change
  - Prototyping
  - Incremental Delivery
  - Boehm's spiral model

2.3 Coping with change



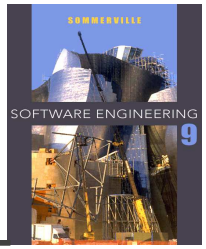
- Change inevitable in all large software projects.
- Rework expensive
- 2 approaches to reduce cost of rework:

2.3 Coping with change



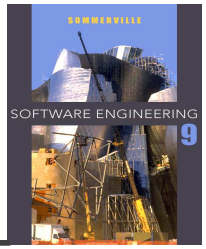
- Change inevitable in all large software projects.
- Rework expensive
- 2 approaches to reduce cost of rework:
  - **Change avoidance**
    - E.g. a prototype to refine requirements

2.3 Coping with change



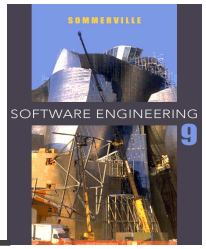
- Change inevitable in all large software projects.
- Rework expensive
- 2 approaches to reduce cost of rework:
  - **Change avoidance**
    - E.g. a prototype to refine requirements
  - **Change tolerance**
    - E.g. incremental developnmet, refactoring, ..

2.3.1 Software prototyping



✧ A prototype is an initial version of a system used to demonstrate concepts and try out design options.

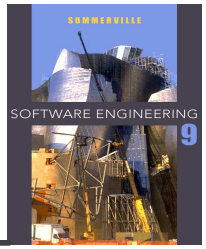
2.3.1 Software prototyping



- A prototype is an initial version of a system used to demonstrate concepts and try out design options.
- A prototype can be used in:
  - Requirements engineering process
  - Design processes

    e.g. database design, UI,

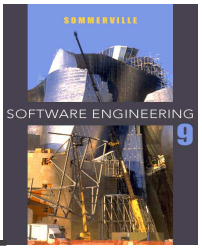
2.3.1 Software prototyping

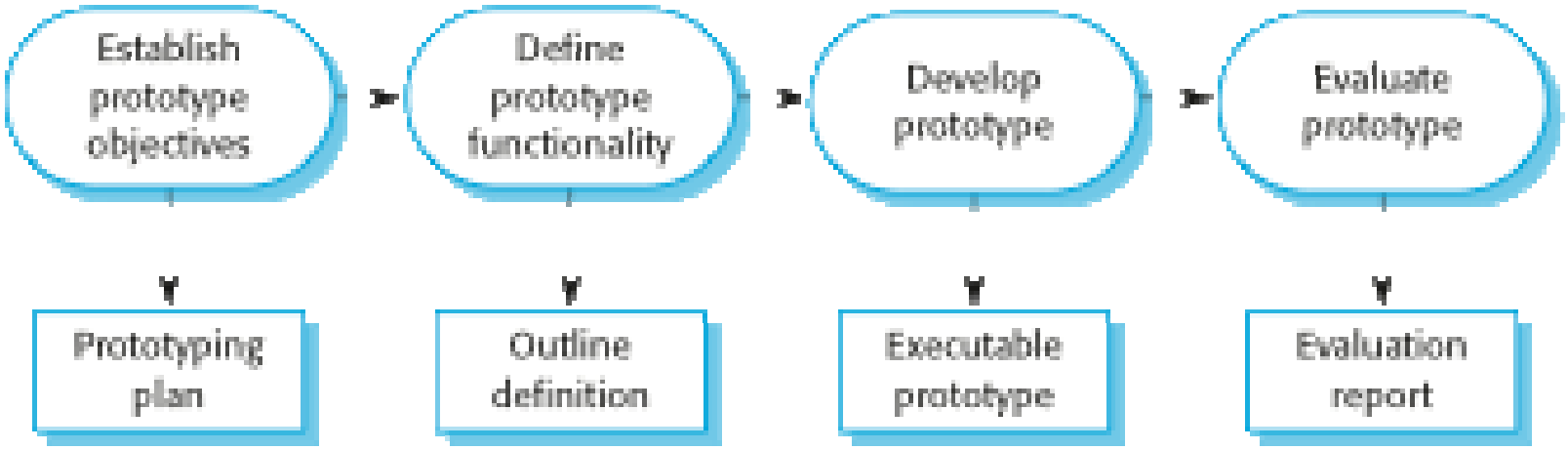


- A prototype is an initial version of a system used to demonstrate concepts and try out design options.
- A prototype can be used in:
  - Requirements engineering process
  - Design processes

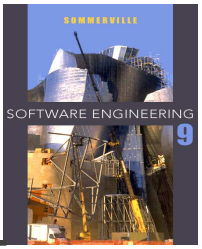
    e.g. database design, UI, ..
- Fast development at low cost

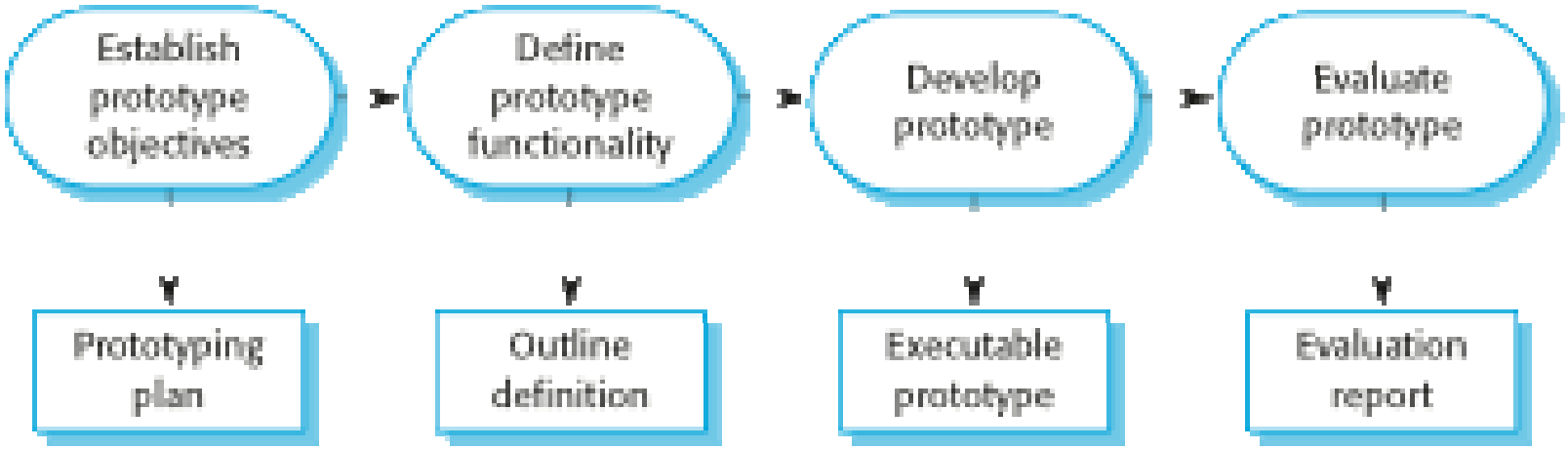
The process of prototype development





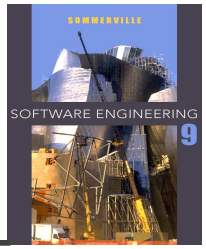
The process of prototype development





Don’t use prototype in final product!

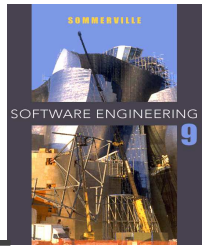
- Relaxed non-functional requirements
- Lack of documentation, degraded system structure ..



✧ Prototypes do not have to be executable to be userful

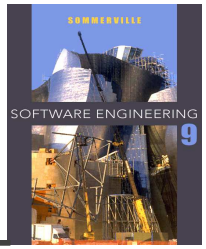
- Paper-based mock-up
- Wizard-of-Oz prototype (only UI developed)
  user interacts with UI, person provides appropriate response

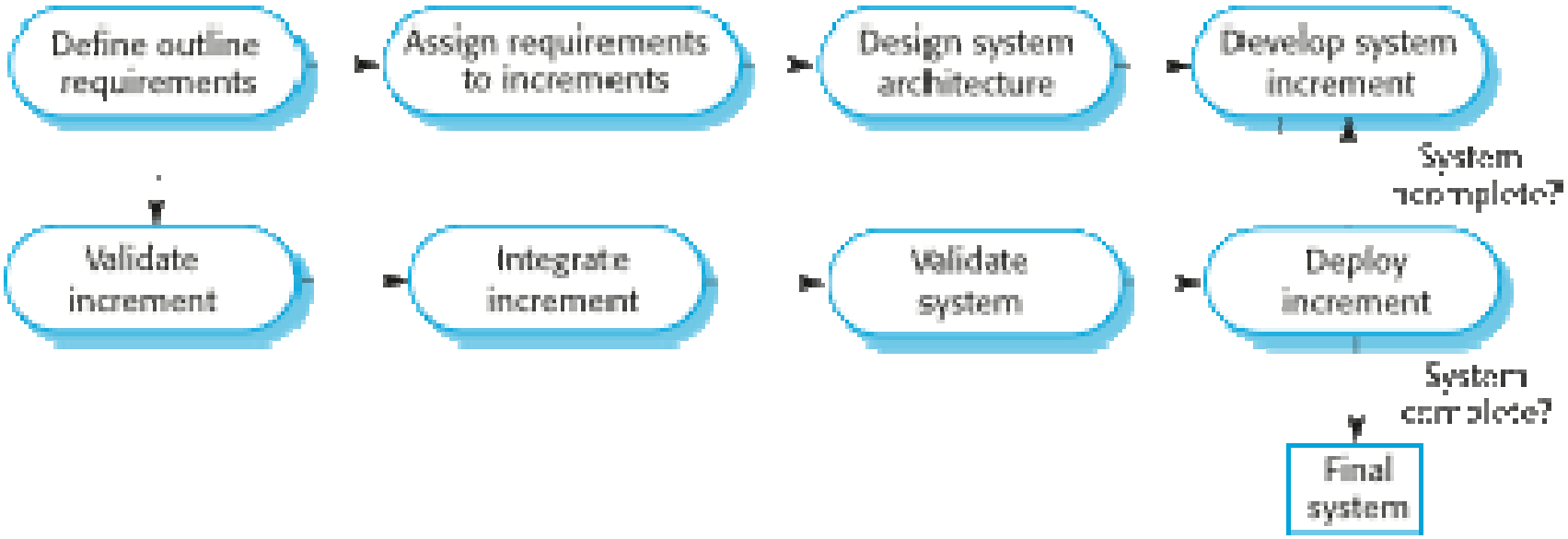
2.3.2 Incremental delivery



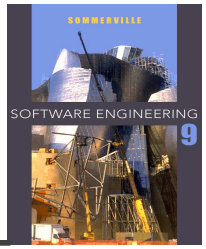
- Development and delivery is broken down into increments with each increment delivering part of the required functionality.
- User requirements prioritised
- Once development of an increment is started, requirements no longer change

Incremental delivery





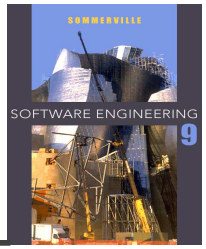
Pro / Con



✧ Pro:

- Customer gains value early on
- Early increments help elicit requirements
- Highest priority components tend to get most testing
- Less likely to fail, easier to adapt to change

Pro / Con



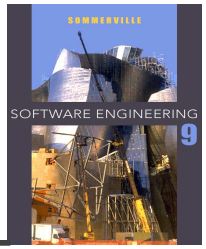
✧ Pro:

- Customer gains value early on
- Early increments help elicit requirements
- Highest priority components tend to get most testing
- Less likely to fail, easier to adapt to change

✧ Con:

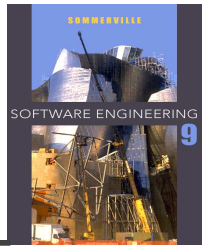
- Harder to identify common facilities used by all increments
- Replacement systems => difficulties with users (want all)
- No complete requirements up front => difficulties w. managers

2.3.3 Boehm's spiral model



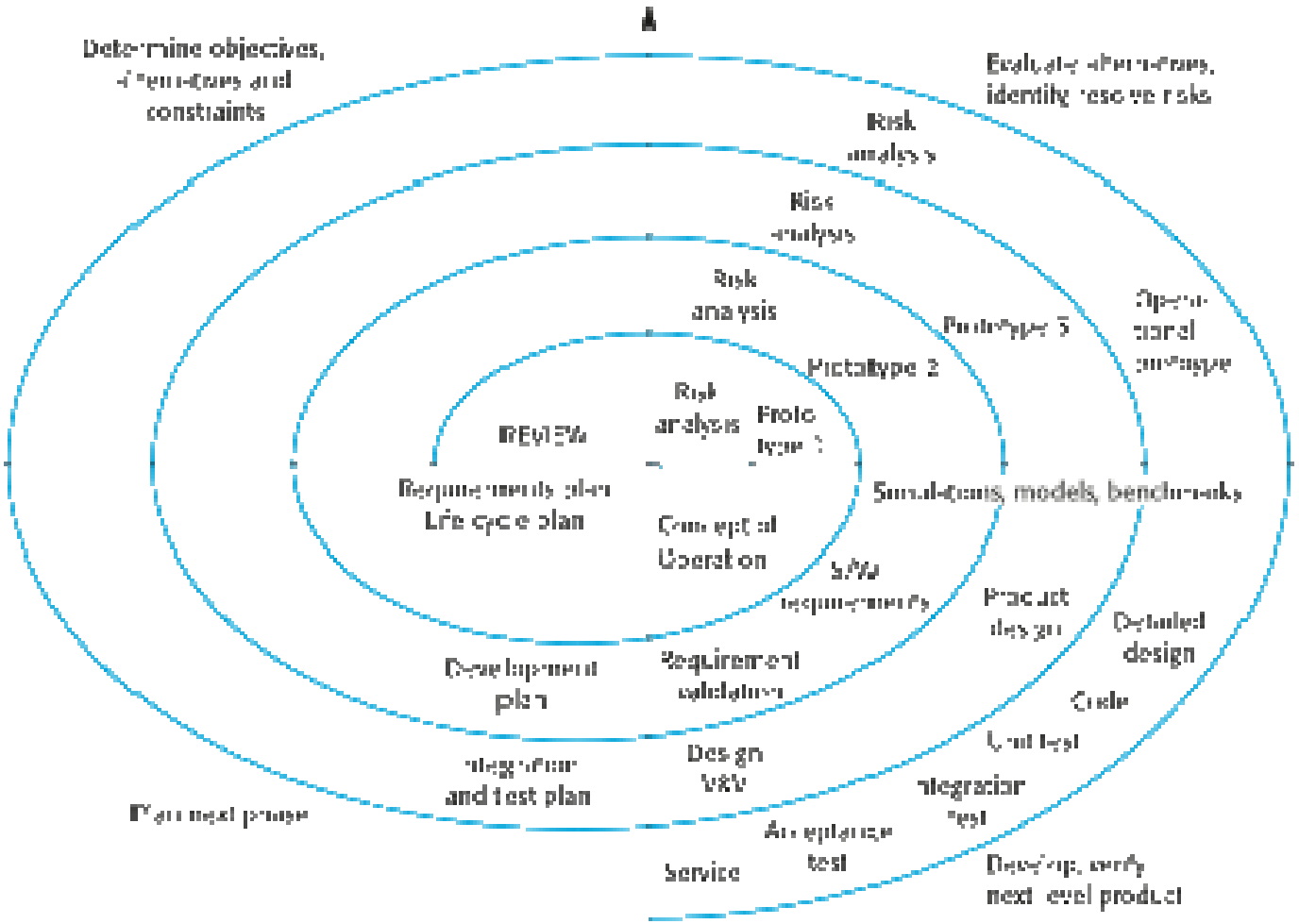
- Risk-driven software process framework
- Risks are explicitly assessed and resolved throughout the process.
- Each loop in the spiral represents a phase in the process.
- No fixed phases such as specification or design - loops in the spiral are chosen depending on what is required

2.3.3 Boehm's spiral model

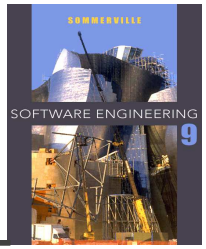


- Risk-driven software process framework
- Risks are explicitly assessed and resolved throughout the process.
- Each loop in the spiral represents a phase in the process.
- No fixed phases such as specification or design - loops in the spiral are chosen depending on what is required.
- Change avoidance and change tolerance

Boehm's spiral model of the software process



Spiral model usage



- Spiral model has been very influential in helping people think about iteration in software processes and introducing the risk-driven approach to development.
- In practice, however, the model is rarely used as published for practical software development.