Software Processes

Chapter 2

CMPT 276
© Dr. B. Fraser
Based on slides from Software Engineering 9th ed, Sommerville.

Topics

1) What activities are part of software development
2) What are software process models?

Process Activities

The software process

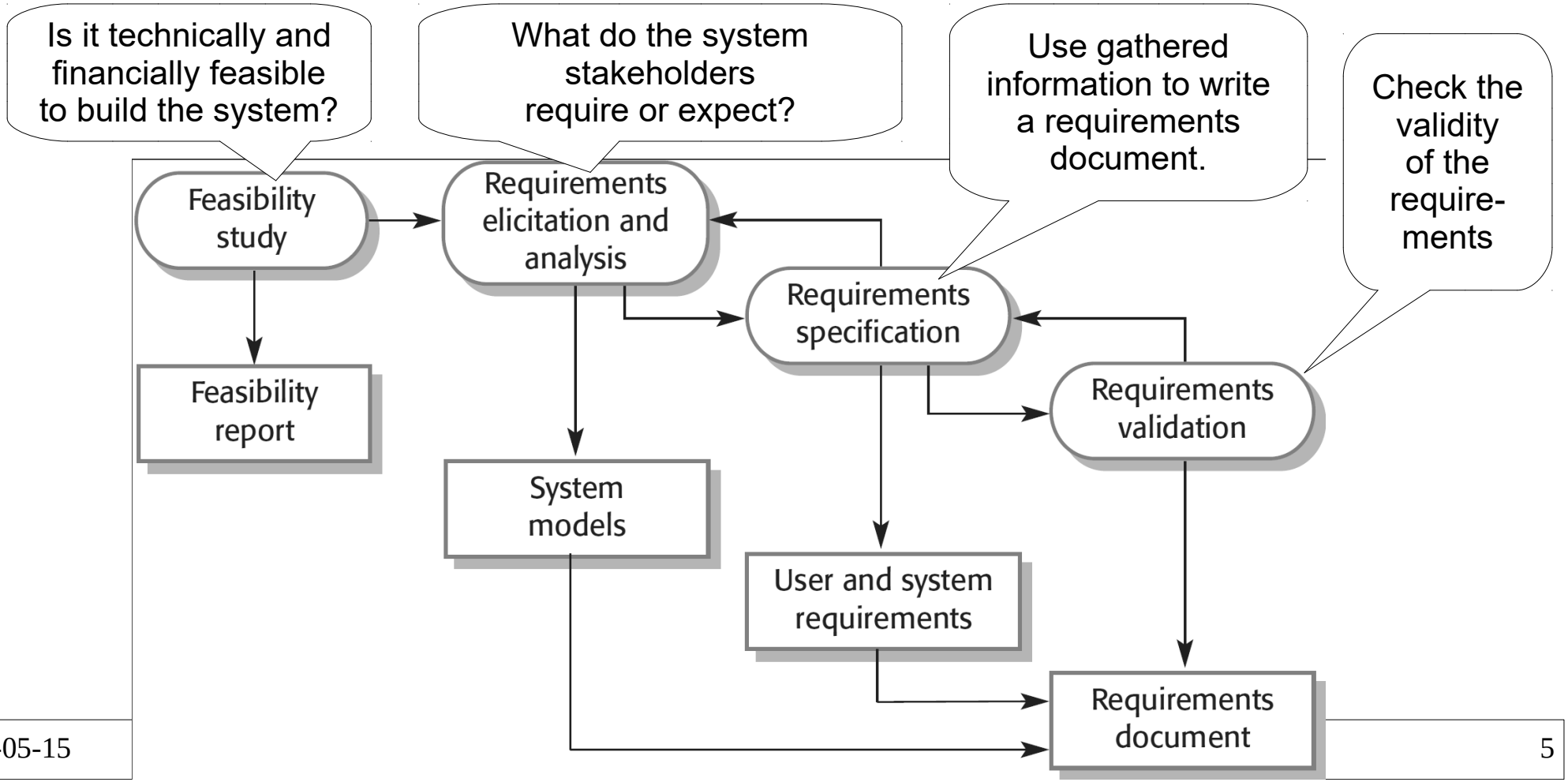
- Software Process:
  - **a structured set of activities required to develop software system**

- All software processes involve:
  - Specification – what will the system do?
  - Design & implementation – how will it do this? ..
    Actually make it work
  - Validation – does it do what the customer wants?
  - Evolution – change system to meet customer's changing needs.
- A software process model is.. **an abstract representation of a real process**

Software Specification

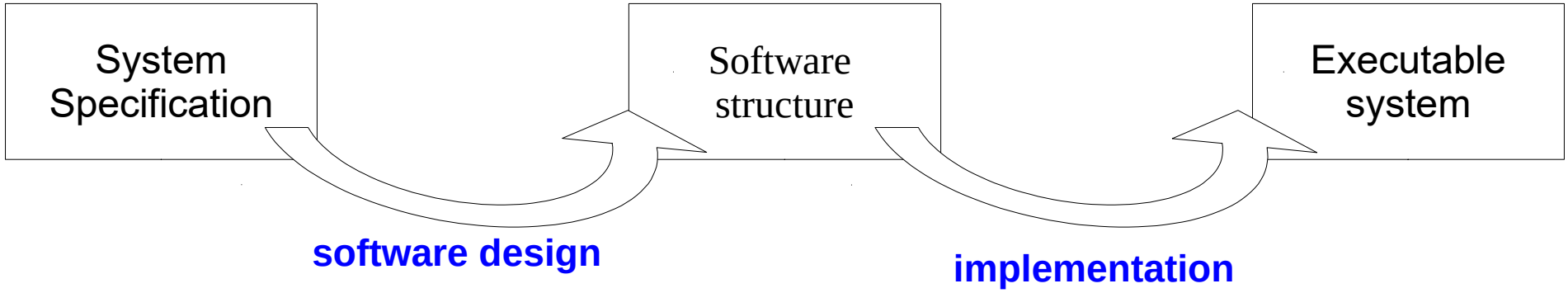
Software specification: establishing what services are required and..

constraints on the system's operation and development (disk size for game)e



Software design and implementation

- Process to convert system specification into an executable system.



- Design and implementation are closely related and..

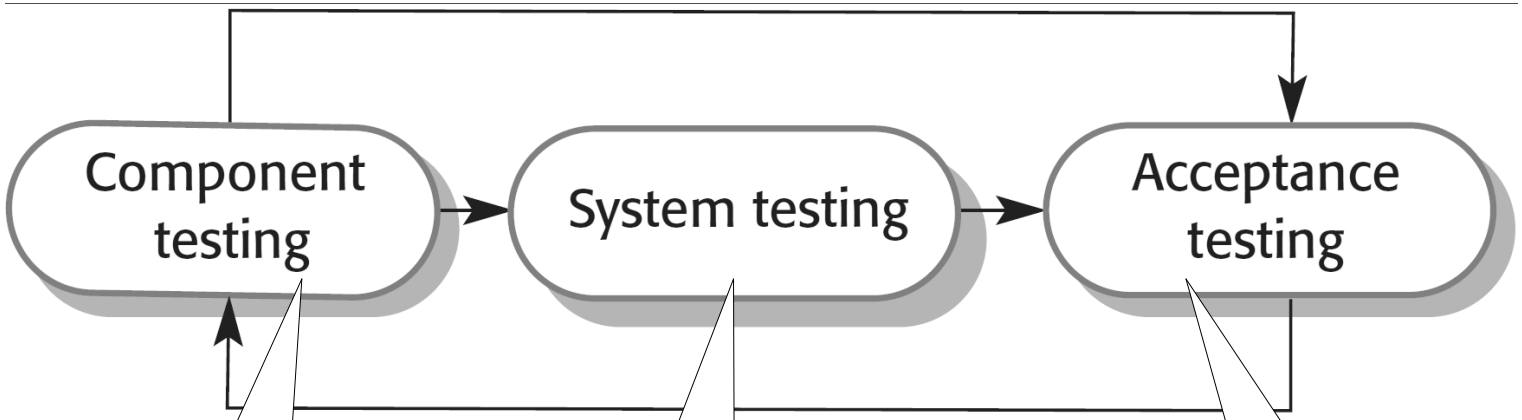
interleaved

Design Activity	Description
Architectural Design	Identify overall structure of the system & principle components:... **subsystems or modules**
UI design	Layout initial ideas for user interface (UI).
Component design	Design each system component
Database design	Design the system's data structures and database

Software validation

- Validation
  - checks the system conforms to its.. **specification and meets customer requirements**
- Involves testing
  - Create test cases which ensure system behaves correctly for some component/feature.
  - Best if using real-world data **tesla auto-drive feature testing**
- Can Involve Formal Verification
  - .. **proving that a system operates correctly**
  - Hard in practice; often restricted to critical components of life-critical components.

Testing Stages



Individual functions or objects are..

tested independently

May test coherent groupings of objects.

Testing of system..

as a whole

Testing of emergent properties is particularly important.

Testing with..

customer data

to check that system meets customer's needs.

Software evolution

- Software is inherently flexible and can change.
- Software must change to meet new business needs
  - Most of a project's time and cost associated with...

    maintenance
- Programming stereotype is:
  - development is creative and interesting, but
  - maintenance is dull.
  - This is increasingly irrelevant as most..
  - Line between old and new is blurring.

So, what's the process to develop software?

Software Processes

Software processes

- Describe each process by:
  - **activities in the process** such as designing how data is stored, or the user interface, etc
  - **ordering of these activies**
- All processes involve the four basic activities
  - specification, development, validation and evolution.
- 2 Big Questions
  - **planning** Done up front? Or as you go?
  - **delivery** Done at the end? Or multiple times?

(Planning) Paradigms

- Plan-driven processes:
  - **all process activities planned in advance**
  - **progress measured against this plan**
  - Also called Big Design Up Front (BDUF).
- Agile processes:
  - **planning is incremental**
  - Easier to change the process to reflect changing customer requirements.
- Most practical processes include elements of both plan-driven and agile approaches.
  - **There is no right or wrong software processes**

Delivery

- Single Delivery (at end)
  - Software only delivered to customer.. **once it is fully completed**

- Incremental Delivery
  - Customer is given.. **incomplete versions of software through development** of the software throughout development.

High-level View of Software Processes

	Single Delivery	Incremental Delivery
Plan Driven (BDUF)	**waterfall**	Plan Driven Incremental Model, Spiral Model
Evolutionary Planning	**none**	**agile= scrum or XP**

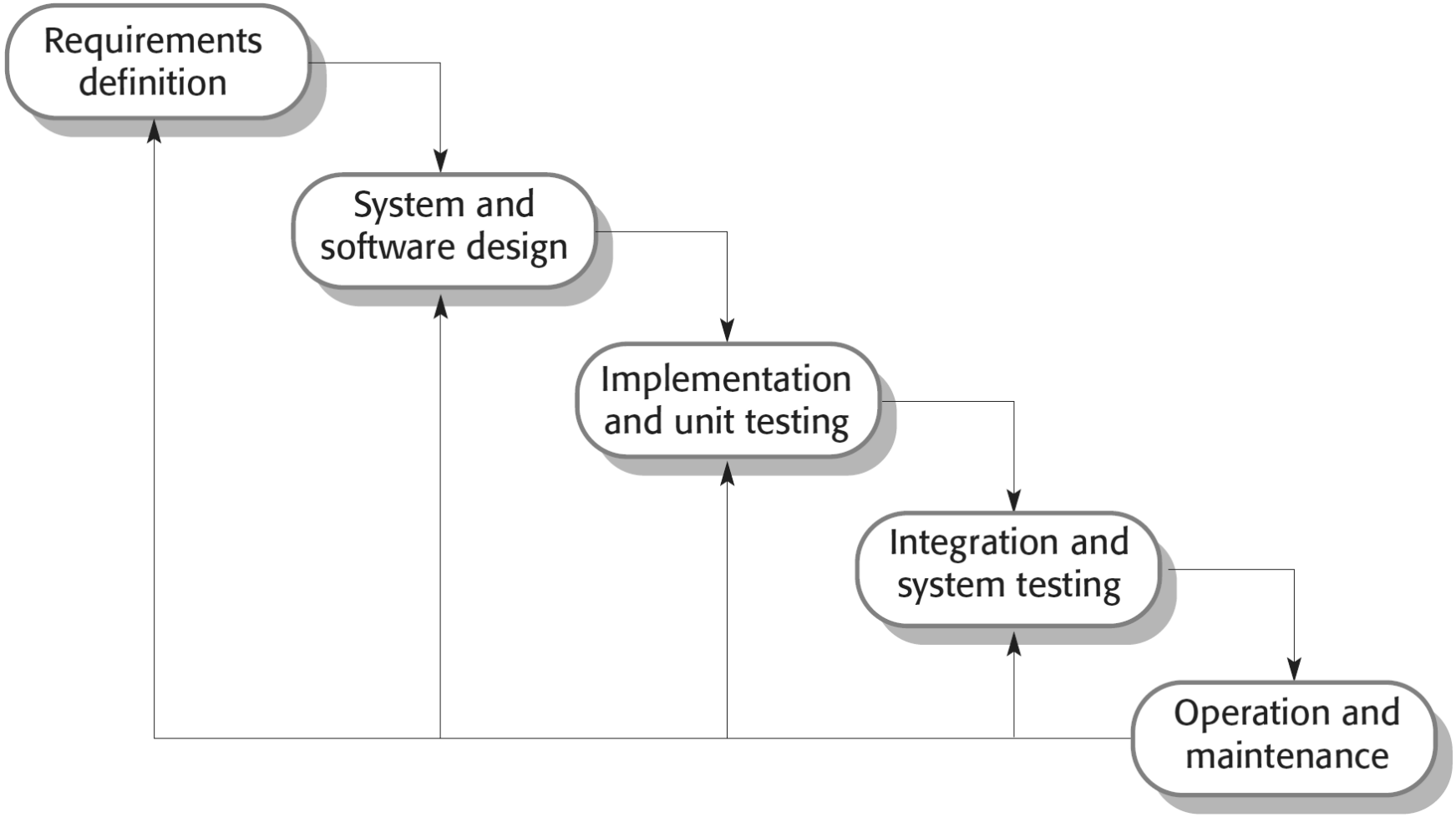
Describe what a course assignment would look like for each of these 4 possibilities.

Software process models

- The waterfall model
  - Plan-driven model – Separate and distinct phases of specification and development.
- Incremental development
  - Specification, development and validation are..

    interleaved
- Agile
  - Lightweight process to adapt to changing requirements.
- Most large systems developed using a process that incorporates elements from multiple models.

Waterfall model phases



Waterfall model problems

inflexible stages make it difficult to meet changing customer requirements

- Must complete phase N before starting phase N+1.
- Waterfall model is (somewhat) appropriate when..

  requirements are well-understood and change is limited

  - Few business systems have stable requirements.
- Plan-driven nature of the waterfall model helps..

  coordinate the work

  - However waterfall is so rigid it is virtually never used as a full methodology.

"Walking on water and developing software from a specification are easy if..

both are frozen

-- Edward Berard (1993)

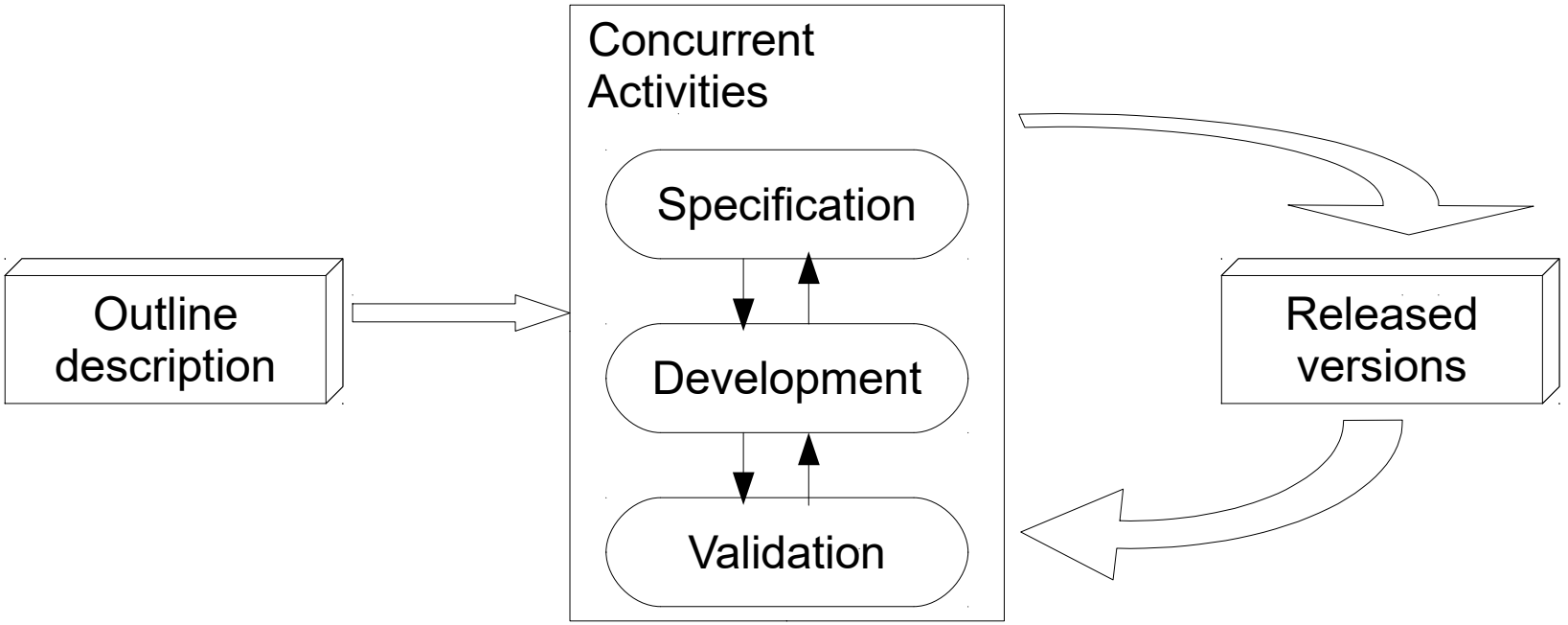
Incremental development

- Waterfall model delivers full system to user..

  at the end of the process

- Incremental development delivers..

  incomplete intermediate versions



Incremental and its benefits

- Incremental development usable by either paradigm
  - Plan Driven Models:
    Functionality of increments are.. **planned in advance**
  - Agile Models:
    Functionality of early increments are planned,
    later increments driven by... **customer needs**
- Reduced cost from changing customer requirements:
  - Not as much.. **code written that must change**
- Quick delivery of useful software.
  - Easier to get customer feedback on working software rather than paper designs.
  - Customer uses and gains value from the software earlier than with a single end delivery process.

Incremental development problems

- Code Rot:
  - **regular changes tend to corrupt system's structure**

    Incorporating code changes becomes increasingly difficult and costly.
  - Time and money must be spent refactoring to improve the software.

Refactoring

- Refactoring
  - **improving code without adding new features**
- Refactoring Examples
  - rename a poorly named variable
  - split huge function into smaller ones,
  - improve OOD (object oriented design)
  - fixing parts of the code which have..

    poor code quality or poor readability

Agile

- Agile methodologies are lightweight:
  they try to.. **reduce process overhead**
  - Ex: Only as much documentation and planning as needed.
- Develop application in short iterations
  - ~1-3 weeks long
  - .. **select features** at start of each iteration.
  - .. **deliver working software** at end of each iteration.
- Very common in industry
  - Whole slide-deck on it soon!

Summary

- Software processes are the activities involved in producing a software system.
  - Requirements engineering: develop the specification.
  - Design and implementation: transform requirements specification into an executable software system.
  - Software validation: check the system conforms to its specification and meets the needs of its users.
  - Software evolution: change existing software systems to meet new requirements.
- Process models describe a sequence of activities: ‘waterfall’ model, incremental development, and agile development.