EECS 2311

Software Development Project

Click to edit Master text styles
Second level
Third level
Fifth level

January 6, 2015

YORK UNIVERSITÉ UNIVERSITY

Reading

- Author: Ian Sommerville
- Title: Software Engineering
- Chapters 1-4

Software Engineering

- Software Engineering is the science and art of building significant software systems that are:
  1. on time
  2. on budget
  3. with acceptable performance
  4. with correct operation

Software Engineering

- The economies of all developed nations are dependent on software.
- More and more systems are software controlled.
- Software engineering is concerned with theories, methods and tools for professional software development.
- Software engineering expenditure represents a significant fraction of the GNP of developed countries.

Software Costs

- Software costs often dominate system costs. The costs of software on a PC are often greater than the hardware cost.
- Software costs more to maintain than it does to develop.
- Software engineering is concerned with cost-effective software development.

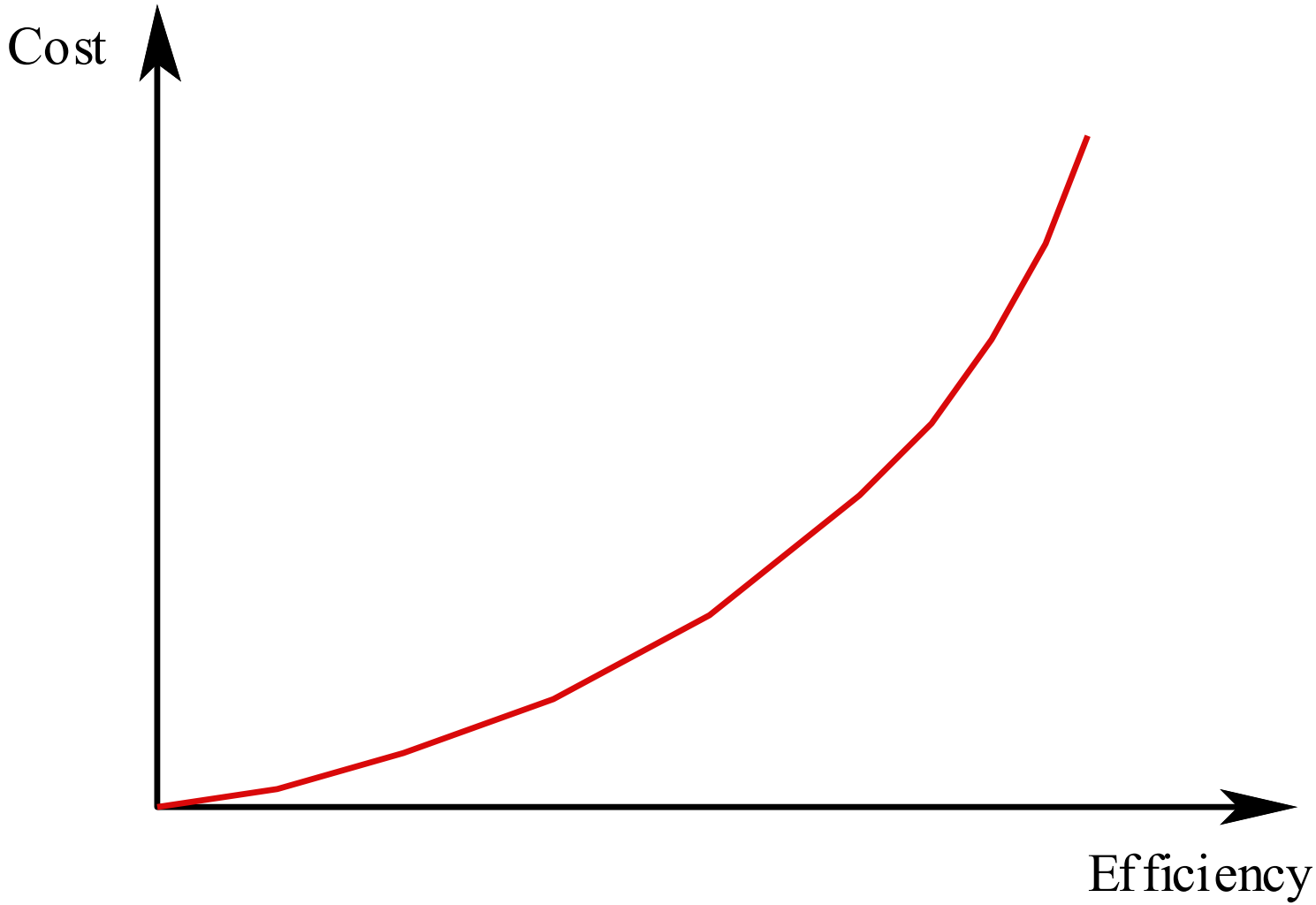
Software Product Attributes

- Maintainability
- Dependability
- Efficiency
- Usability

Importance of Product Characteristics

- The relative importance of these characteristics depends on the product and the environment in which it is to be used.
- In some cases, some attributes may dominate
  - In safety-critical real-time systems, key attributes may be dependability and efficiency.
- Costs tend to rise exponentially if very high levels of any one attribute are required.

Efficiency Costs



The Software Process

- Structured set of activities required to develop a software system
  - Specification
  - Design
  - Validation
  - Evolution
- Activities vary depending on the organization and the type of system being developed.

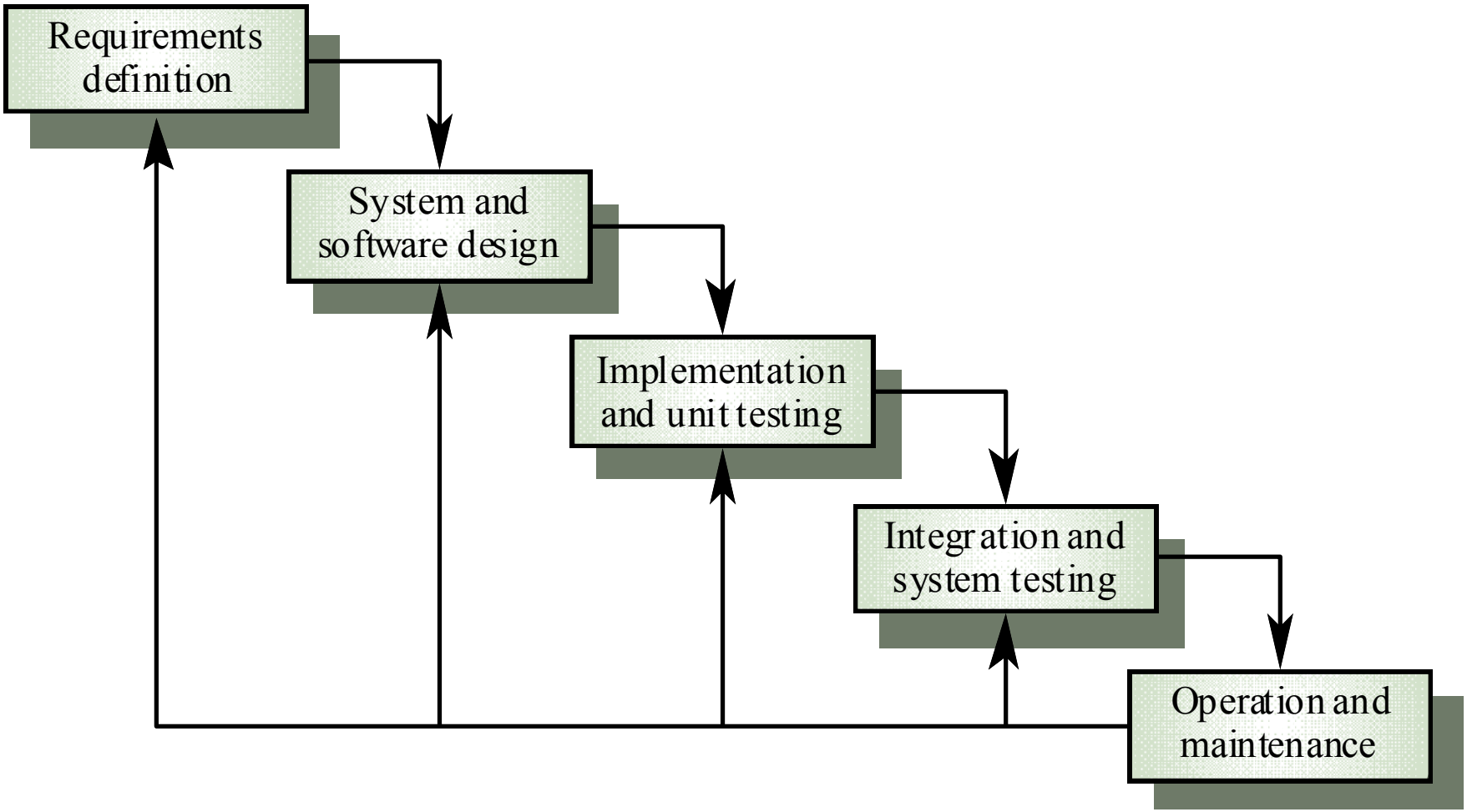
Engineering Process Model

- **Specify**: Set out the requirements and constraints on the system.
- **Design**: Produce a model of the system.
- **Manufacture**: Build the system.
- **Test**: Check the system meets the required specifications.
- **Install**: Deliver the system to the customer and ensure it is operational.
- **Maintain**: Repair faults in the system as they are discovered.

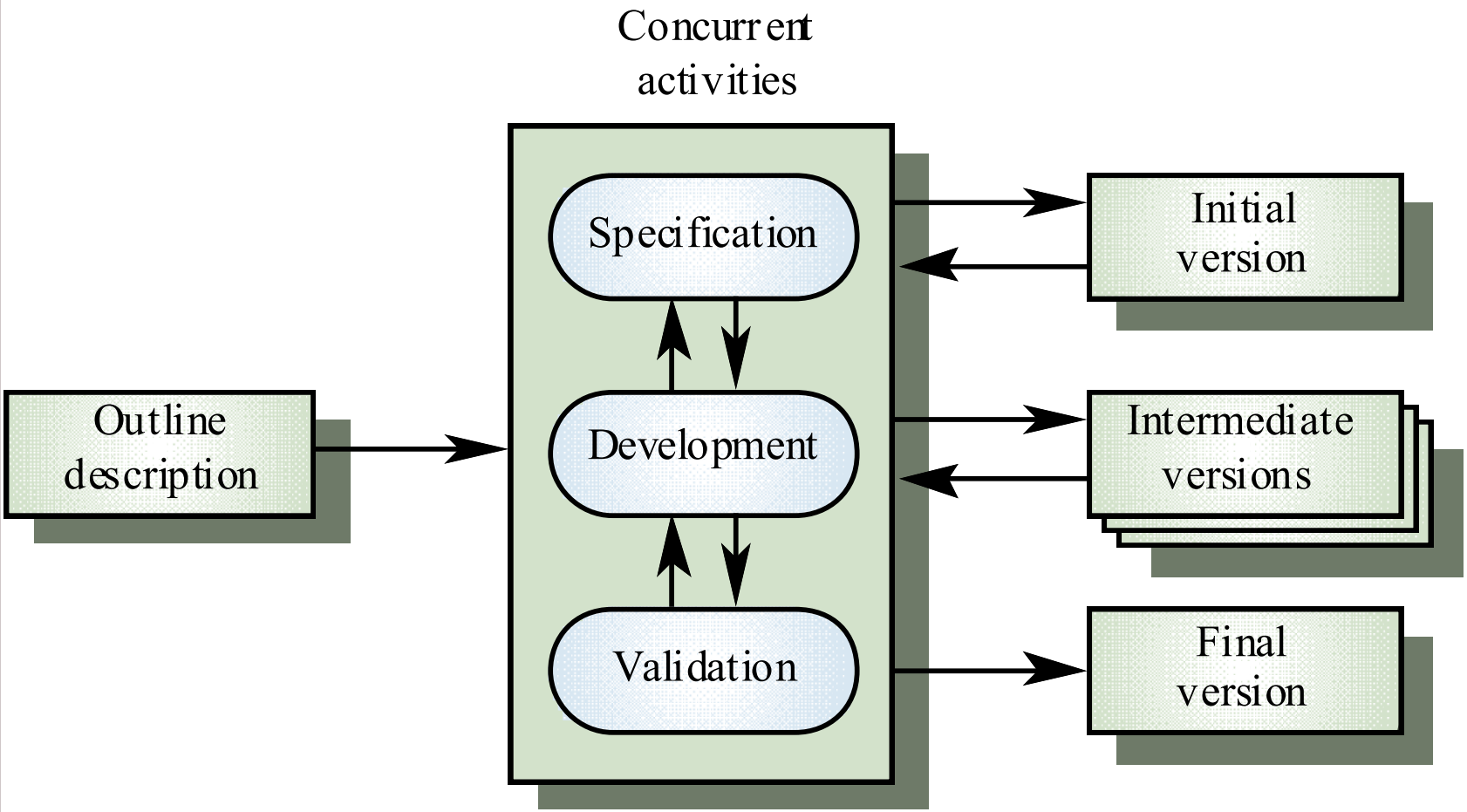
Software Engineering is Different

- Normally, specifications are incomplete.
- Very blurred distinction between specification, design and manufacture.
- No physical realization of the system for testing.
- Software does not wear out - maintenance does not mean component replacement.

Waterfall Process Model



Evolutionary Process Model



Agile methods

- Dissatisfaction with the overheads involved in design methods led to the creation of agile methods. These methods:
  - Focus on the code rather than the design;
  - Are based on an iterative approach to software development;
  - Are intended to deliver working software quickly and evolve this quickly to meet changing requirements.
- Agile methods are probably best suited to small/ medium-sized business systems or PC products.

Principles of agile methods

- Customer involvement
- Small releases
- Embrace change
- Test-first design
- Refactoring
- Continuous integration

Our project

- Fully develop a system that translates guitar tablature from ASCII to PDF
- We will call it TAB2PDF
- We will use an agile approach whenever possible

ASCII Tablature

```
TITLE=Moonlight Sonata
SUBTITLE=Ludwig van Beethoven
SPACING=5

|-------------------------|-------------------------|
|-----1-----1-----1-----1-|-----1-----1-----1-----1-|
|---2-----2-----2-----2---|---2-----2-----2-----2---|
|-2-----2-----2-----2-----|-2-----2-----2-----2-----|
|-0-------------------------|-------------------------|
|-------------------------|-3-----------------------|

|-------------------------|-------------------------|
|-----1-----1-----3-----3-|-----3-----1-----0-----0-|
|---2-----2-----3-----3---|---1-----2-----2-----1---|
|-3-----3-----3-----3-----|-2-----2-----2-----0-----|
|-------------5-----------|-------------------------|
|-1-----------------------|-0-----------0-----------|
```

PDF Tablature

To get started

- Look at the sample input and output files posted on the course website
- Download the iText library for dynamically creating PDF files
  - http://itextpdf.com
- Attempt to create a Hello World PDF file
- See the examples posted on the course webpage

Intentionally vague requirements

- In a real software development project, requirements are vague and ever-changing
- The exact requirements will be refined iteratively by meeting with the “customer” on a weekly basis

Teams

- Teams are assigned randomly by the “manager”
- As enrollment in the course changes in the first few weeks, the “manager” will rearrange the teams
- Same as a real software project!

Team 1

- Atto, Brody
- Cirillo, Marco
- Patel, Deep
- Ragavendran, Varsha
- Sitiugin, Glib

Team 2

- Abou-Nassar, Rami
- Arindaeng, Kevin
- James, Abasifreke
- McVicar, Daniel
- Sebthosseini, Behshad

Team 3

- El Shafie, Adham
- Ismail, Yahya
- Sharma, Nisha
- Tang, Darren
- Zaki, Mina

Team 4

- Chauhan, Yash
- Mule, Ayrton
- Polakkattil, Albin
- To, Jennifer
- Vaisman, Edward
- Zhang, Yingying

Workload

- This course requires 8-10 hours per week per student
- Have to start working immediately
- In the second part of each lecture, each team will present their progress to the instructor and receive feedback
  - “Customer” on site!

Evaluation

- 20% - Midterm prototype + Presentation (due Feb 25)
- 80% - Final project + Presentation (due Apr 1)

- Each team submission will receive a grade based on its merit. Individual grades may be less if full participation has not been demonstrated.