Department Of Computer Science

Software Engineering Concepts (CSC291)

Lecture 01

Introduction

Topics Covered

- **Professional software development**
  - What is meant by software engineering.
  - FAQs about software engineering

- To **introduce software engineering** and to explain its importance

- To set out the **answers to key questions about software engineering**

FAQs on Software Engineering (I)

- What is **software**?
- What is **software engineering**?
- What is the **difference between software engineering and computer science?**
- What is the **difference between software engineering and system engineering**?
- What is a **software process**?
- What is a **software process model**?

FAQs on Software Engineering (I)

- What are the **costs of software engineering**?
- What are **software engineering methods**?
- What are the **attributes of good software**?
- What are the **key challenges facing software engineering**?
- What are ***Stakeholders in Software Engineering***?
- What are **Difficulties and Risks in Software Engineering**?

What is Software?

- **Software:** **“Computer programs and associated documentation such as requirements, design models and user manuals”.**

- Software products may be developed for a particular customer or may be developed for a general market.

Software

- **Program:** The program or code itself is definitely included in the software.
- **Data:** The data on which the program operates is also considered as part of the software.
- **Documentation:** Another very important thing that most of us forget is documentation. All the documents related to the software are also considered as part of the software.
- So the software is not just the code written in Cobol, Java, Fortran or C++. It also includes the data and all the documentation related to the program.

Types of Software.

- **Custom**
  - For a specific customer
- **Generic**
  - Sold on open market
  - Often called
    - **COTS** (Commercial Off The Shelf)
    - **Shrink-wrapped** (Refers to store-bought software, implying a standard platform that is widely supported.)
- **Embedded**
  - Built into hardware
  - Hard to change

Types of Software

- **Differences among custom, generic and embedded software**

	Custom	Generic	Embedded
Number of *copies* in use	low	medium	high
Total *processing power* devoted to running this type of software	low	high	medium
Worldwide annual *development effort*	high	medium	low

Types of Software

- **Real time software**
  - **Example**. Control and monitoring systems
  - Must react immediately
  - Safety often a concern
- **Data processing software**
  - Used to run businesses
  - Accuracy and security of data are key
- ***Some software have both aspects***

Software Products

- **Generic Products**
  - Stand-alone systems that are marketed and sold to any customer who wishes to buy them.
  - **Examples** – PC software such as graphics programs, project management tools; CAD software;
- **Customized Products**
  - Software that is commissioned by a specific customer to meet their own needs.
  - **Examples** – embedded control systems, air traffic control software, traffic monitoring systems.

Product Specification

- **Generic products**
  - The specification of what the software should do is **owned by the software developer and decisions** on software change **are made by the developer**.

- **Customized products**
  - The specification of what the software should do is **owned by the customer for the software and they make decisions on software** changes that are required.

Software Applications

- 1. **System software:** such as compilers, editors, file management utilities
- 2. **Application software**: stand-alone programs for specific needs.
- 3.**Engineering/scientific software**: Characterized by "number crunching"algorithms. such as automotive stress analysis, molecular biology, orbital dynamics etc
- 4. **Embedded software resid**es within a product or system. (key pad control of a microwave oven, digital function of dashboard display in a car)
- 5. **Product-line software** focus on a limited marketplace to address mass consumer market. (word processing, graphics, database management)
- 6. **WebApps** (Web applications) network centric software. As web 2.0 emerges, more sophisticated computing environments is supported integrated with remote database and business applications.
- 7. **AI software** uses non-numerical algorithm to solve complex problem. Robotics, expert system, pattern recognition game playing

Changing Nature of Software

- **Four broad categories of software** are evolving to dominate the industry
  - Web Applications
  - Mobile Applications
  - Cloud Computing
  - Product Line Software

What is Software Engineering?

- **Definition:**
- "Software engineering is an engineering discipline that is concerned with all aspects of software production".

- **Engineering discipline**
  - Using appropriate theories and methods to solve problems bearing in mind organizational and financial constraints.
- **All aspects of software production**
  - Not just technical process of development. Also project management and the development of tools, methods etc. to support software production.

Software Engineering Definition

The seminal definition:

[**Software engineering is**] the establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines.

The IEEE definition:

Software Engineering: The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.

Software engineers should adopt a **systematic and organised approach** to their work and use **appropriate tools and techniques** depending on the problem to be solved, the development constraints and the resources available.

Computer Science Vs. Software Engineering

"Software Engineering is the process of utilizing our knowledge of computer science in effective production of software systems."

Computer Science	Software Engineering
Computer science is concerned with the theories and methods that underlie computers and software system.	Software engineering is concerned with the practical problems of producing software.

What is the difference between Software Engineering and Computer Science?

- "**Computer science is concerned** with theory and fundamentals; while software engineering is concerned with the practicalities of developing and delivering useful software".

- **Computer science theories are still insufficient** to act as a complete underpinning (supporting) for software engineering (unlike e.g. physics and electrical engineering).

What is the difference between Software Engineering and System Engineering?

- "**System engineering is concerned with** all aspects of computer-based systems development including hardware, software and process engineering. While Software engineering is part of this process concerned with developing the software infrastructure, control, applications and databases in the system".

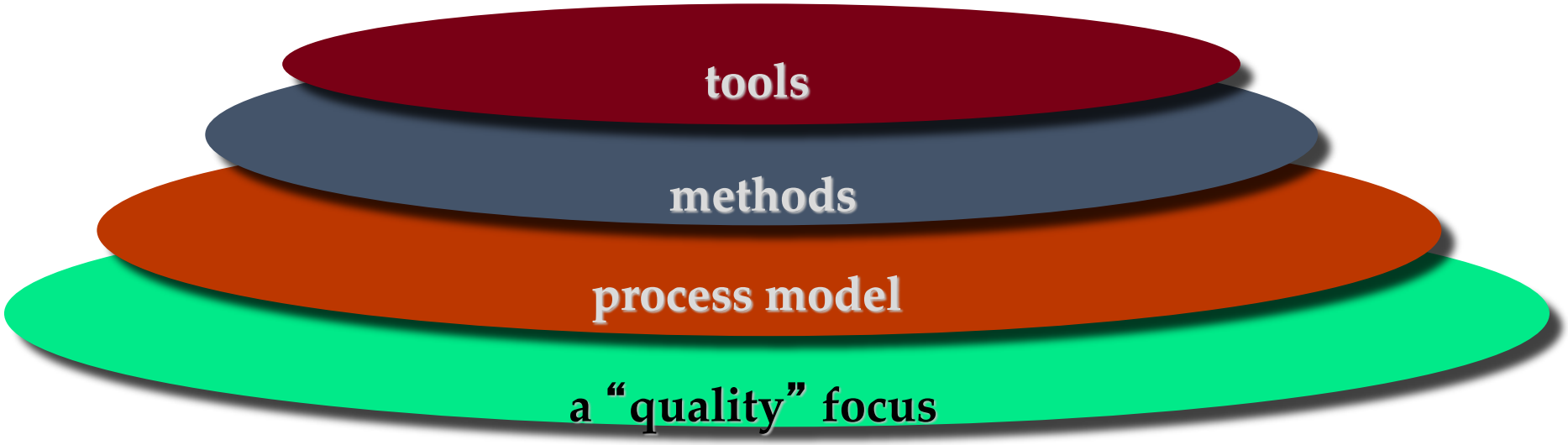
- **System engineers are involved** in system specification, architectural design, integration and deployment.

➢ System Engineering & Software Engineering

System Engineering	Software Engineering
System engineering is concerned with all aspects of the development and evolution of complex systems where software plays a major role.	Software engineering is concerned with the practical problems of producing software.
System engineering is therefore concerned with hardware development, policy and process design and system deployment, as well as software engineering.	
System engineers are involved in specifying the system, defining its overall architecture, and then integrating the different parts to create the finished system.	

Software Engineering

A Layered Technology



Software Engineering
A Layered Technology

- Any engineering approach must rest on organizational commitment to **quality** which fosters a continuous process improvement culture.
- **Process layer as the foundation defines a framework** with activities for effective delivery of software engineering technology. Establish the context where products (model, data, report, and forms) are produced, milestone are established, quality is ensured, and change is managed.
- **Method provides technical how-to's for building software**. It encompasses many tasks including communication, requirement analysis, design modeling, program construction, testing and support.
- **Tools** provide automated or semi-automated support for the process and methods.

A Software Engineering Framework

- **A quality focus**
  - Any engineering approach must rest on organizational commitment to quality which fosters a continuous process improvement culture.
- **Process layer**
  - The foundation for software engineering is the process layer
  - **Defines a framework** that must be established for effective delivery of software engineering technology
  - **Establish a context where**
    - Products (model, data, report, and forms) are produced
    - Milestones are established
    - Quality is ensured
    - Change is managed

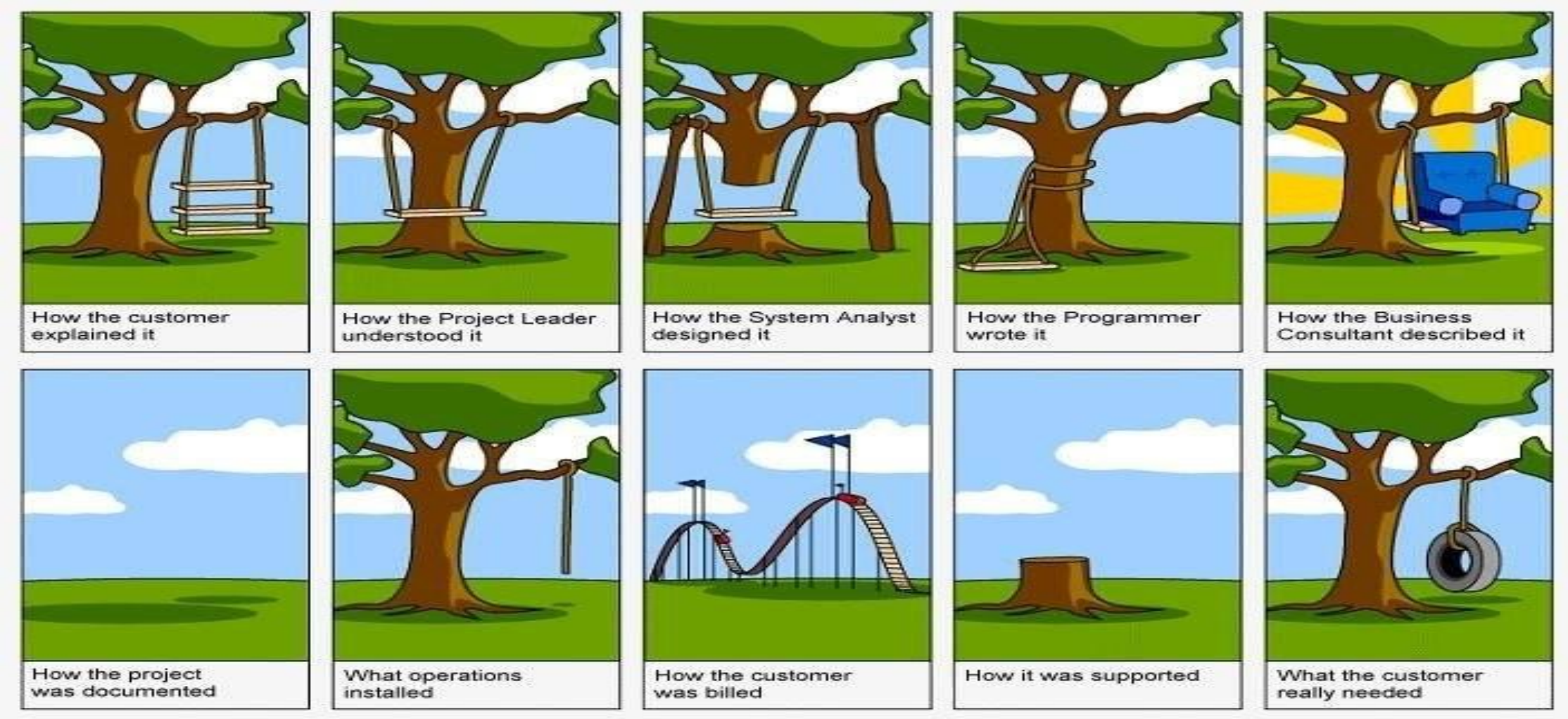
A Software Engineering Framework

- **Methods**
  - Provide the technical how-to for building software
  - **Methods encompass a broad array of tasks**
    - Communication , requirements analysis, design modeling, program construction, testing, and support
- **Tools**
  - Provide automated or semi-automated support for the process and the methods
  - **Integrated tools to support software development**
    - Called *computer aided software engineering*

Importance of Software Engineering

- More and more, **individuals and society rely on advanced software systems**. We need to be able to produce reliable and trustworthy systems economically and quickly.
- **It is usually cheaper**, in the long run, to use software engineering methods and techniques for software systems rather than just write the programs as if it was a personal programming project. **For most types of system,** the majority of costs are the costs of changing the software after it has gone into use.

Importance of Software Engineering



What is a Software Process?

- Software Process: "The systematic approach that is used in software engineering is sometimes called a software process."

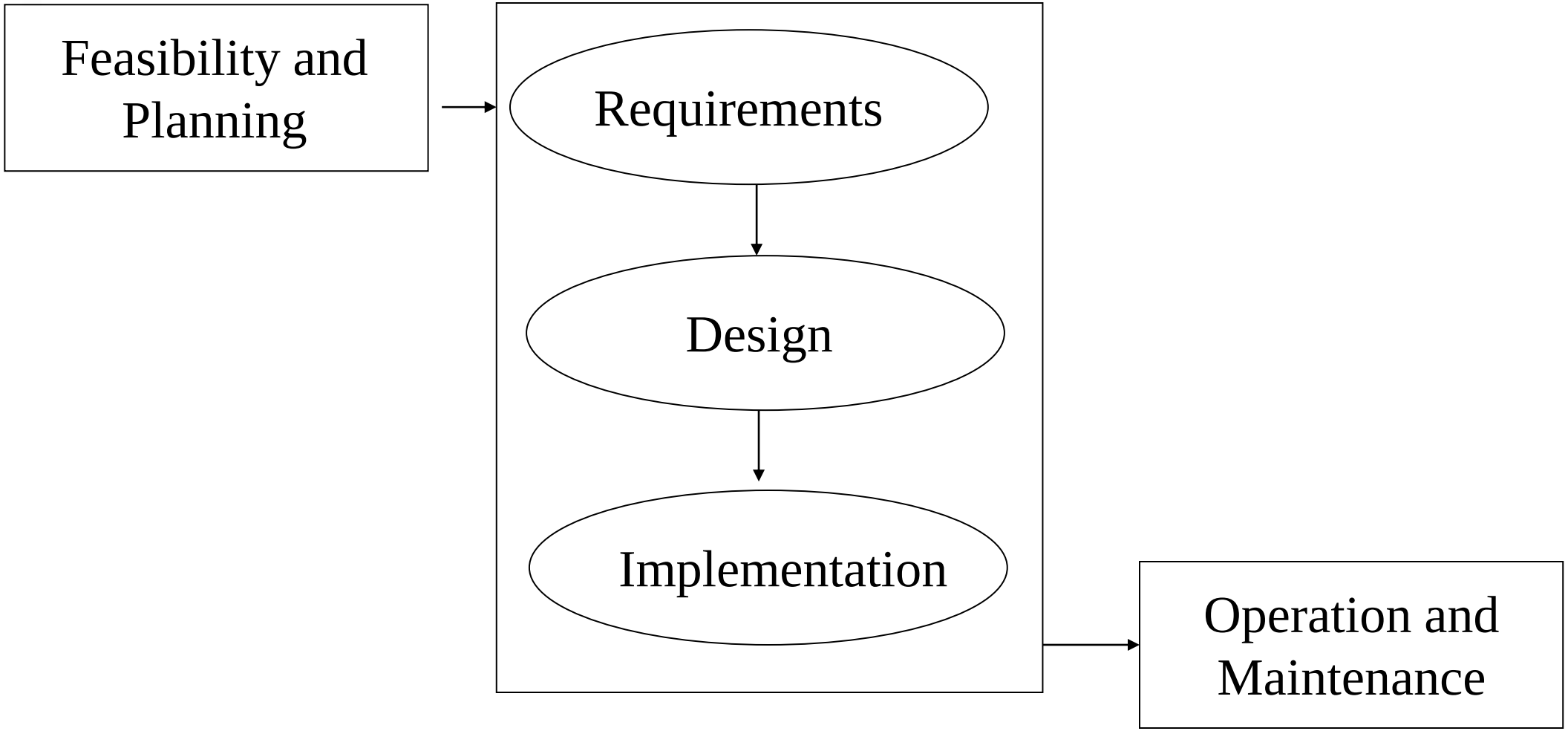
- "A set of activities whose goal is the development or evolution of software".

- Generic activities in all software processes are:
  - **Software specification**
  - **Software development:**
  - **Software validation:**
  - **Software evolution:**

Activities in Software Process?

- Generic activities in all software processes are:
  - **Software Specification** - what the system should do and its development constraints
    - where customers and engineers define the software that is to be produced and the constraints on its operation
  - **Software Development** - production of the software system
    - where the software is designed and programmed
  - **Software Validation** - checking that the software is what the customer wants
    - where the software is checked to ensure that it is what the customer requires.
  - **Software Evolution** - changing the software in response to changing demands
    - where the software is modified to reflect changing customer and market requirements.

The Software Process (Simplified)



What is a Software Process Model?

- Software Process Model: "A simplified representation of a software process, presented from a specific perspective".

- **Examples of process perspectives are**:
  - **Work flow** ➔ what is done when? - sequence of activities;
  - **Data flow** ➔ which information flows where?
  - **Role / action** ➔ who does what?
- **Generic process models**
  - Waterfall
  - Iterative development
  - Component-based software engineering.

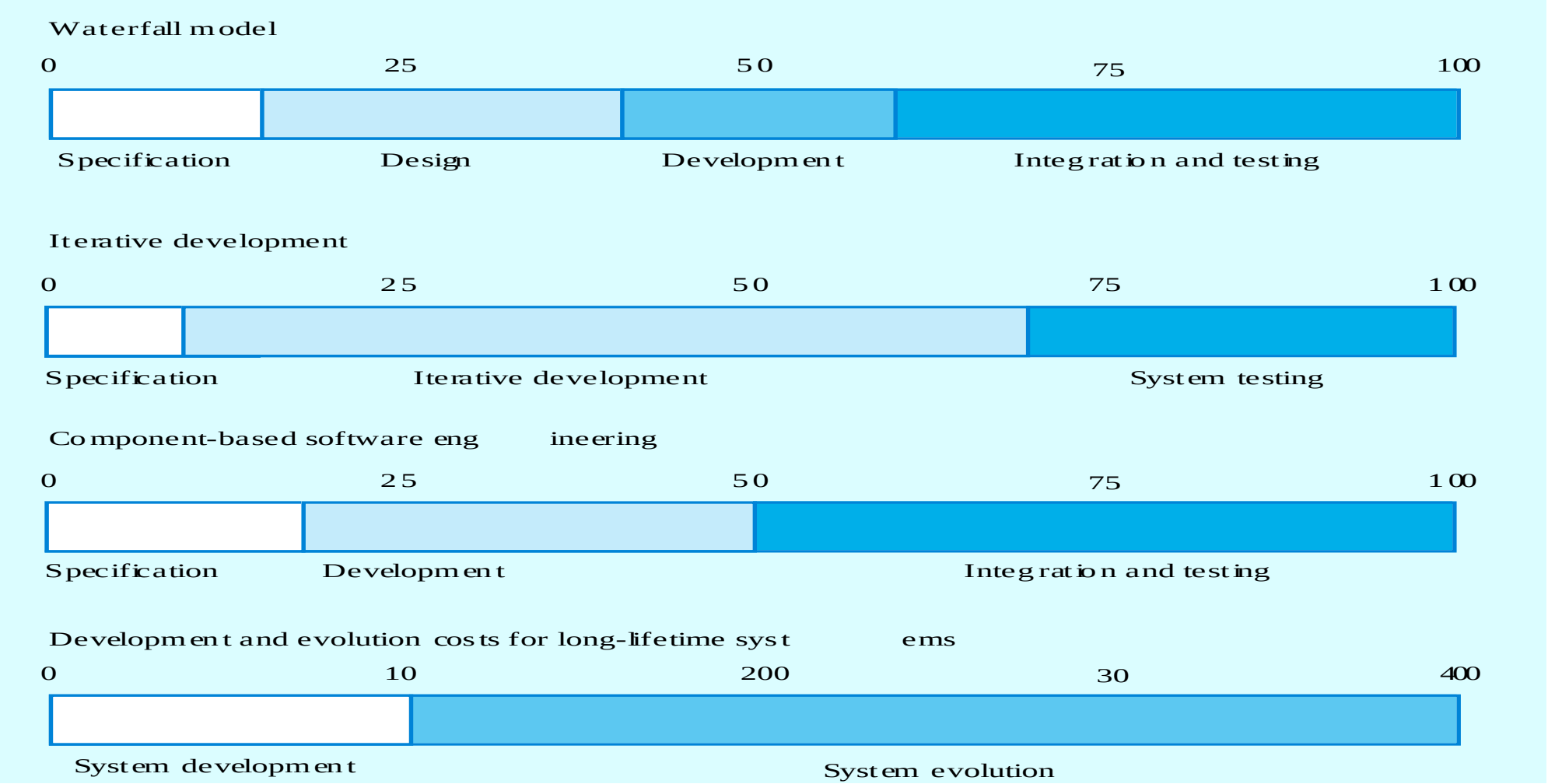
What are the costs of software engineering?

- **Roughly 60% of costs are development costs, 40% are testing costs**. For custom software, evolution costs often exceed development costs.

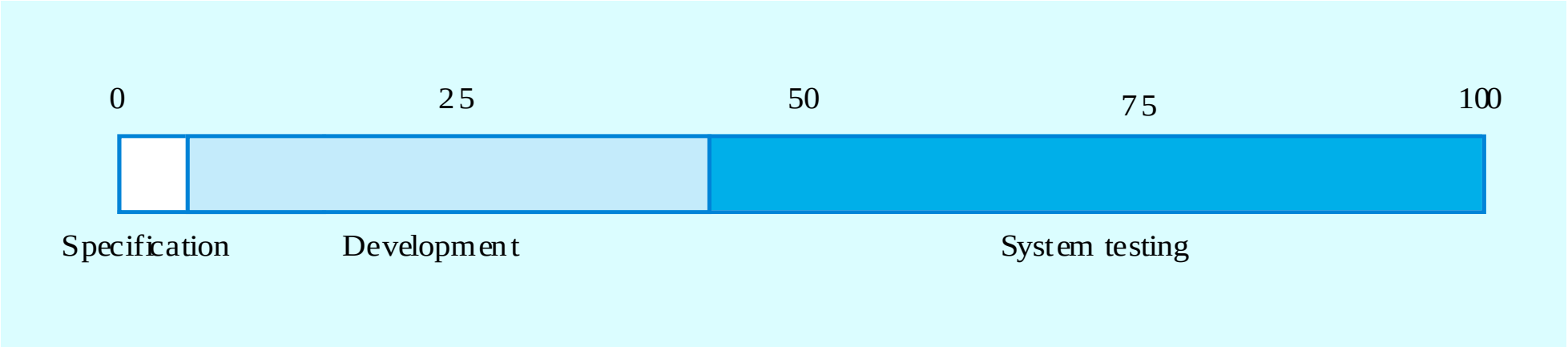
- **Costs vary depending on the type of system** being developed and the requirements of system attributes such as performance and system reliability.

- **Distribution of costs depends** on the development model that is used.

Activity Cost Distribution



Product Development Costs



What are Software Engineering Methods?

- Software Engineering Methods: *"**Structured approaches to software development** which include system models, notations, rules, design advice and process guidance".*
- **Model descriptions**
  - Descriptions of graphical models which should be produced
- **Rules**
  - Constraints applied to system models
- **Recommendations**
  - Advice on good design practice
- **Process guidance**
  - What activities to follow

What is CASE?
(Computer-Aided Software Engineering)

- CASE: "Software systems that are intended to provide automated support for software process activities".
- **CASE systems are often used for method support.**
- **Upper-CASE**
  - Tools to support the **early process activities of requirements and design;**
- **Lower-CASE**
  - Tools to support **later activities such as programming, debugging and testing.**

What are the attributes of good Software?

- **The software should deliver the required functionality** and performance to the user and should be maintainable, dependable and acceptable.

- **Maintainability**
  - Software must evolve to meet changing needs
- **Dependability**
  - Software must be trustworthy
- **Efficiency**
  - Software should not make wasteful use of system resources
- **Acceptability**
  - Software must accepted by the users for which it was designed. This means it must be understandable, usable and compatible with other systems.

What are the key challenges facing Software Engineering?

- **Key challenges are:** Heterogeneity, delivery, trust and *Legacy systems*
- **Heterogeneity**
  - Developing techniques for building software that can cope with heterogeneous platforms and execution environments.
  - Large-scale systems are often distributed and include a mix of hardware and software
- **Delivery**
  - Developing techniques that lead to faster delivery of software.
  - There is increasing pressure for ever faster delivery of software
- **Trust**
  - Developing techniques that demonstrate that software can be trusted by its users.
- **Legacy systems**
  - Valuable existing systems must be maintained and updated

What are Stakeholders in Software Engineering?

- **Stakeholder:** A stakeholder is anyone who has a stake in the successful outcome of the project
- **1. Users**
  - Those who use the software
- **2. Customers**
  - Those who pay for the software
- **3. Software developers**
- **4. Development Managers**

Note: All four roles can be fulfilled by the same person

What are Difficulties and Risks in Software Engineering?

- **Complexity** and large numbers of details
- Uncertainty **about technology**
- Uncertainty **about requirements**
- Uncertainty about **software engineering skills**
- **Constant change**
- **Political risks**

Software Engineering Fundamentals

- Some fundamental principles apply to all types of software system, irrespective of the development techniques used:
  - **Systems should be developed** using a managed and understood development process. Of course, different processes are used for different types of software.
  - **Dependability and performan**ce are important for all types of system.
  - **Understanding and managing the software** specification and requirements (what the software should do) are important.
  - **Where appropriate**, **you should reuse software** that has already been developed rather than write new software.

Web Software Engineering

- **The Web is now a platform** for running application and organizations are increasingly developing web-based systems rather than local systems
- **Software reuse is** the dominant approach for constructing web-based systems.
- **When building these systems**, you think about how you can assemble them from pre-existing software components and systems.
- **Web-based systems should be developed and delivered incrementally.**
  - It is now generally recognized that it is impractical to specify all the requirements for such systems in advance.
- **User interfaces are constrained by the capabilities of web browsers**.
  - Technologies such as AJAX allow rich interfaces to be created within a web browser but are still difficult to use. Web forms with local scripting are more commonly used.

Web-based Software Engineering

- **Web-based systems are complex** distributed systems but the fundamental principles of software engineering discussed previously are as applicable to them as they are to any other types of system.

- **The fundamental ideas of software engineering**, discussed in the previous section, apply to web-based software in the same way that they apply to other types of software system.

Key Points - Frequently asked questions about software engineering

Question	Answer
What is software?	Computer programs and associated documentation. Software products may be developed for a particular customer or may be developed for a general market.
What are the attributes of good software?	Good software should deliver the required functionality and performance to the user and should be maintainable, dependable and usable.
What is software engineering?	Software engineering is an engineering discipline that is concerned with all aspects of software production.
What are the fundamental software engineering activities?	Software specification, software development, software validation and software evolution.
What is the difference between software engineering and computer science?	Computer science focuses on theory and fundamentals; software engineering is concerned with the practicalities of developing and delivering useful software.
What is the difference between software engineering and system engineering?	System engineering is concerned with all aspects of computer-based systems development including hardware, software and process engineering. Software engineering is part of this more general process.

Key Points - Frequently asked questions about software engineering

Question	Answer
What are the key challenges facing software engineering?	Coping with increasing diversity, demands for reduced delivery times and developing trustworthy software.
What are the costs of software engineering?	Roughly 60% of software costs are development costs, 40% are testing costs. For custom software, evolution costs often exceed development costs.
What are the best software engineering techniques and methods?	While all software projects have to be professionally managed and developed, different techniques are appropriate for different types of system. For example, games should always be developed using a series of prototypes whereas safety critical control systems require a complete and analyzable specification to be developed. You can't, therefore, say that one method is better than another.
What differences has the web made to software engineering?	The web has led to the availability of software services and the possibility of developing highly distributed service-based systems. Web-based systems development has led to important advances in programming languages and software reuse.

Essential attributes of good software

Product characteristic	Description
Maintainability	Software should be written in such a way so that it can evolve to meet the changing needs of customers. This is a critical attribute because software change is an inevitable requirement of a changing business environment.
Dependability and security	Software dependability includes a range of characteristics including reliability, security and safety. Dependable software should not cause physical or economic damage in the event of system failure. Malicious users should not be able to access or damage the system.
Efficiency	Software should not make wasteful use of system resources such as memory and processor cycles. Efficiency therefore includes responsiveness, processing time, memory utilisation, etc.
Acceptability	Software must be acceptable to the type of users for which it is designed. This means that it must be understandable, usable and compatible with other systems that they use.

Key Points

- **Software engineering is an engineering discipline** that is concerned with all aspects of software production.
- **Essential software product attributes are** maintainability, dependability and security, efficiency and acceptability.
- **The high-level activities of** specification, development, validation and evolution are part of all software processes.
- **The fundamental notions of software engineering** are universally applicable to all types of system development.
- **There are many different types of system** and each requires appropriate software engineering tools and techniques for their development.
- **The fundamental ideas of software engineering** are applicable to all types of software system.


Thank
you