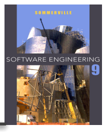
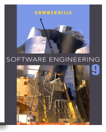


Chapter 1- Introduction

Lecture 1



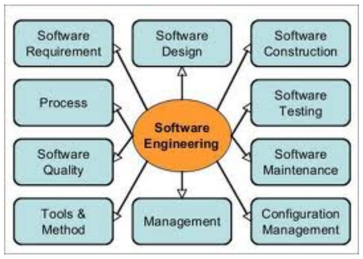
Software engineering

✧ The economies of ALL developed nations are dependent on software.

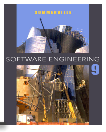
✧ More and more systems are software controlled

✧ Software engineering is concerned with theories, methods and tools for professional software development.

✧ Expenditure on software represents a significant fraction of GNP in all developed countries.



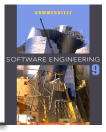
Ingeniería de software



- ✧ Def. IEEE. La aplicación sistemática, disciplinada y medible, orientada al desarrollo, operación y mantenimiento del software, esto es, la aplicación de métodos y prácticas de ingeniería al software.
- ✧ Def. El establecer y utilizar principios de ingeniería para obtener software que es confiable y que funciona eficientemente en equipos de cómputo y dentro parámetros económicos sustentables

3

Ingeniería de Software



Desde 1968 hasta la fecha diferentes organismos (SEI, IEEE, ISO) han enfocado sus esfuerzos en:

- Identificar los factores clave que determinan la calidad del software.
- Identificar los procesos necesarios para producir y mantener software.
- Definir la base de conocimiento necesaria para la producción y mantenimiento de software.

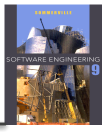
El resultado ha sido la **necesidad de profesionalizar el desarrollo, mantenimiento y operación de los sistemas de software**, introduciendo métodos y formas de trabajo sistemáticos, disciplinados y cuantificables.

4

Fuente: Juan Palacio

Ingeniería de Software:

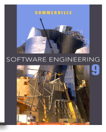
Principales organizaciones de estandarización.



- **ISO: International Organization for Standardization**
- **IEEE- Computer Society: Institute of Electrical and Electronics Engineers**
- **SEI: Software Engineering Institute**

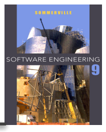
Fuente: Juan Palacio

IEEE software life cycle



- SPM – Software project management **IEEE 1058**
- SRS – Software requirements specification **IEEE 830**
- SRS – Systems and software engineering — Life cycle processes — Requirements engineering **IEEE 29148**
- SDD – Software design description **IEEE 1016**
- SQA – Software quality assurance **IEEE 730**
- SCM – Software configuration management **IEEE 828**
- STD – Software test documentation **IEEE 829**
- V&V – Software verification and validation **IEEE 1012**
- SUD – Software user documentation **IEEE 1063**
- MTTO – Software maintenance **IEEE 1219**

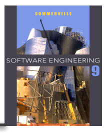
Ingeniería de Software:
Principales organizaciones de estandarización.



La Ingeniería del Software es una ingeniería muy joven que necesitaba:

1. Definirse a sí misma: ¿Cuáles son las áreas de conocimiento que la comprenden?
   - SWEBOK: Software Engineering Body of knowledge
2. Definir los procesos que intervienen en el desarrollo, mantenimiento y operación del software
   - ISO/IEC 12207: Procesos del ciclo de vida del software
3. De las mejores prácticas, extraer modelos de cómo ejecutar esos procesos para evitar los problemas de la "crisis del software"
   - CMM / CMMI
   - ISO/IEC 15504: (SPICE - Software Process Improvement and Capability dEtermination) Marco para métodos de evaluación de procesos.
4. Definir estándares menores para definir criterios unificadores en reqs., pruebas, gestión de la configuración, etc.
   - IEEE 830 - IEEE 1362 - ISO/IEC 14764 …

Ingeniería de Software:
SWEBOK.



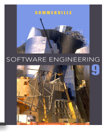
SWEBOK: Necesidad de establecer cuál es el cuerpo de conocimiento que deben conocer los ingenieros del software, y en su desarrollo ha agrupado este conocimiento en **<u>10 áreas</u>**:

1. **Requerimientos**
2. **Diseño**
3. **Construcción**
4. **Pruebas**
5. **Mantenimiento**
6. **Gestión de la configuración**
7. **Gestión de Ing. Sw.**
8. **Procesos de Ing. Sw.**
9. **Herramientas y métodos**
10. **Calidad**

Es importante resaltar que estas áreas no incluyen aspectos clave de las tecnologías de la información, tales como lenguajes específicos de programación, bases de datos, o tecnología de redes y comunicaciones.

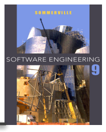
8

Software products



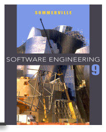
- ✧ Generic products
  - Stand-alone systems that are marketed and sold to any customer who wishes to buy them.
  - Examples – PC software such as graphics programs, project management tools; CAD software; software for specific markets such as appointments systems for dentists.
- ✧ Customized products
  - 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



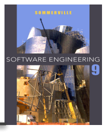
- ✧ Generic products
  - The specification of what the software should do is owned by the software developer and decisions on software change are made by the developer.
- ✧ Customized products
  - The specification of what the software should do is owned by the customer for the software and they make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

Frequently asked questions about software engineering

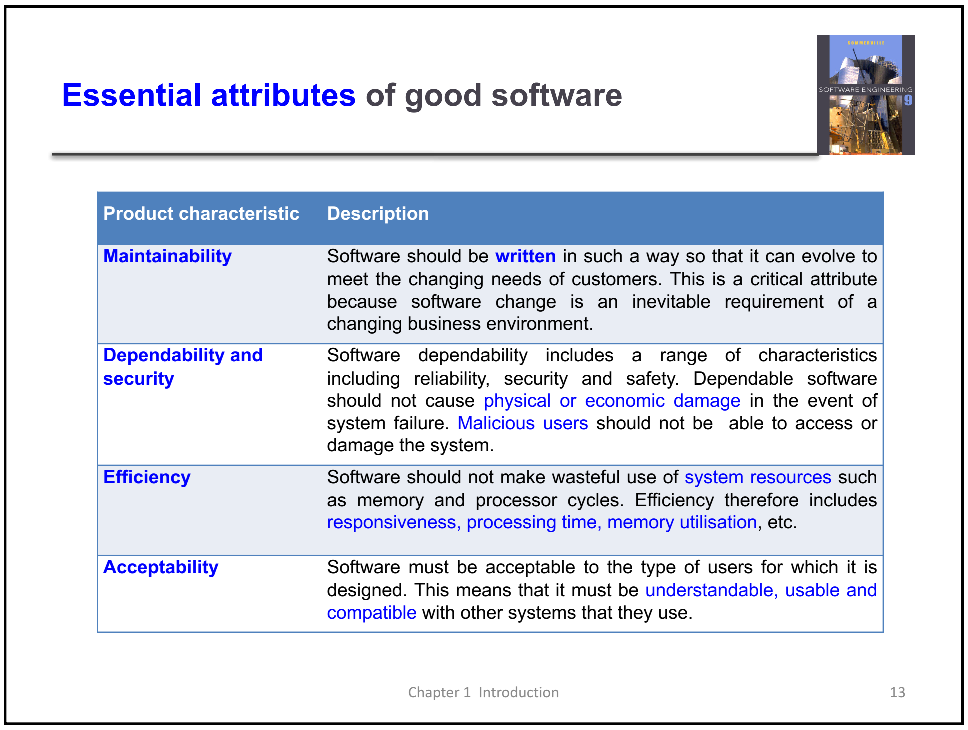


Question	Answer
What is **software**?	**Computer programs and associated documentation**. Software products may be developed for a particular customer or may be developed for a general market.
What are the attributes of **good software**?	Good software should deliver the required **functionality** and 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**.

Frequently asked questions about software engineering



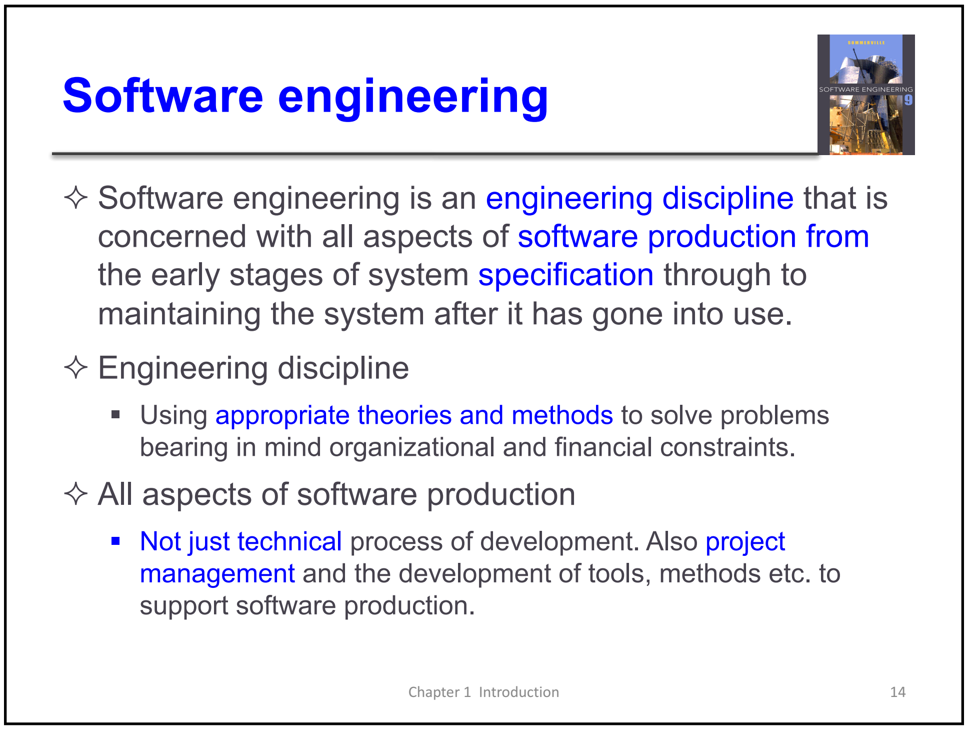
Question	Answer
What are the **key challenges** facing software engineering?	Coping with increasing diversity, demands for **reduced delivery times** and **developing trustworthy software**.
What are the **costs of software engineering**?	Roughly **60%** of software costs are development costs, **40% 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.

Chapter 1 Introduction 13



Software engineering

- ✧ Software engineering is an engineering discipline that is concerned with all aspects of software production from the early stages of system specification through to maintaining the system after it has gone into use.
- ✧ 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.

Chapter 1 Introduction 14

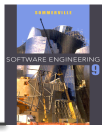
Software process activities

- ✧ Software **specification**, where customers and engineers define the software that is to be produced and the constraints on its operation.
- ✧ Software **development**, where the software is designed and programmed.
- ✧ Software **validation**, where the software is checked to ensure that it is what the customer requires.
- ✧ Software **evolution**, where the software is modified to reflect changing customer and market requirements.

General issues that affect most software

- ✧ Heterogeneity
  - Increasingly, systems are required to operate as distributed systems across networks that include different types of computer and mobile devices.
- ✧ Business and social change
  - Business and society are changing incredibly quickly as emerging economies develop and new technologies become available. They need to be able to change their existing software and to rapidly develop new software.
- ✧ Security and trust
  - As software is intertwined with all aspects of our lives, it is essential that we can trust that software.

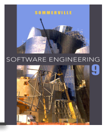
Software engineering fundamentals



✧ 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 performance 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



✧ Software reuse is the dominant approach for constructing web-based systems.

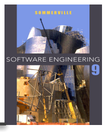
- When building these systems, you think about how you can assemble them from pre-existing software components and systems.

✧ 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.

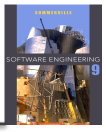
✧ User interfaces are constrained by the capabilities of web browsers.

- 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.



✧The END

This International Standard provides a unified treatment of the processes and products involved in engineering requirements throughout the life cycle of systems and software. This Int. Std is the result of harmonization of the following sources:



- ✧ ISO/IEC 12207:2008 (IEEE Std 12207-2008), *Systems and software engineering — Software life cycle processes*
- ✧ ISO/IEC 15288:2008 (IEEE Std 15288-2008), *Systems and software engineering — System life cycle processes*
- ✧ ISO/IEC/IEEE 15289:2011, *Systems and software engineering — Content of life-cycle information products (documentation)*
- ✧ ISO/IEC TR 19759, *Software Engineering — Guide to the Software Engineering Body of Knowledge (SWEBOK)*
- ✧ **IEEE Std 830, *IEEE Recommended Practice for Software Requirements Specifications*** **IEEE Std 1233, *IEEE Guide for Developing System Requirements Specifications***
- ✧ IEEE Std 1362, *IEEE Guide for Information Technology — System Definition — Concept of Operations (ConOps) Document*
- ✧ ISO/IEC TR 24748-1, *Systems and software engineering — Life cycle management — Part 1: Guide for life cycle management*
- ✧ ISO/IEC/IEEE 24765, *Systems and software engineering — Vocabulary*