

WDMM 3314 Web and Multimedia Engineering
SDEV 2301 Principles of Software Engineering

Chapter 1 Introduction

Part 1 Professional software development

Dr. Nour Almadhoun Alserr

Lecture 2-4

Week 1&2



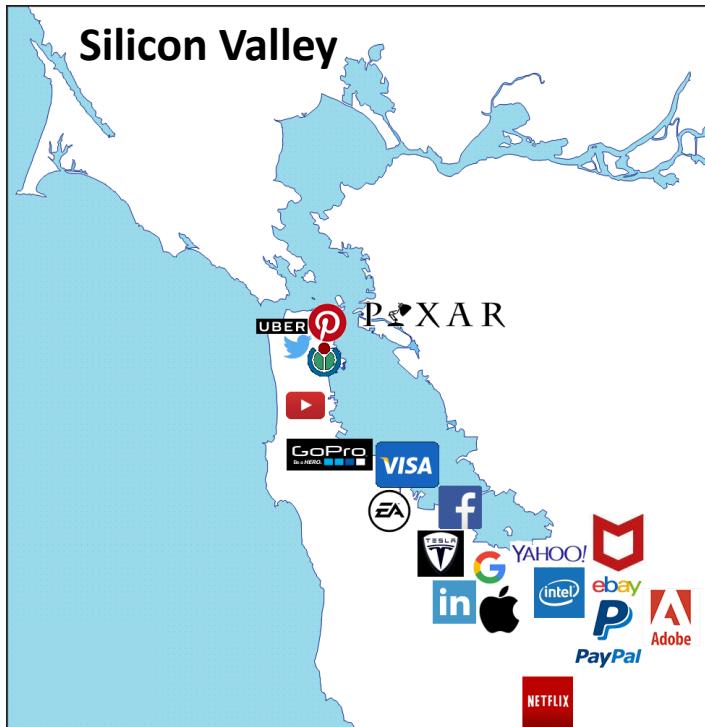


Objectives

- Understand what **software engineering** is and why it is important.
- Understand that the **development of different types of software system** may require different software engineering techniques.

Software Engineering

- The economies of ALL developed nations are dependent on software.
- More and more systems are software controlled.



Software Engineering

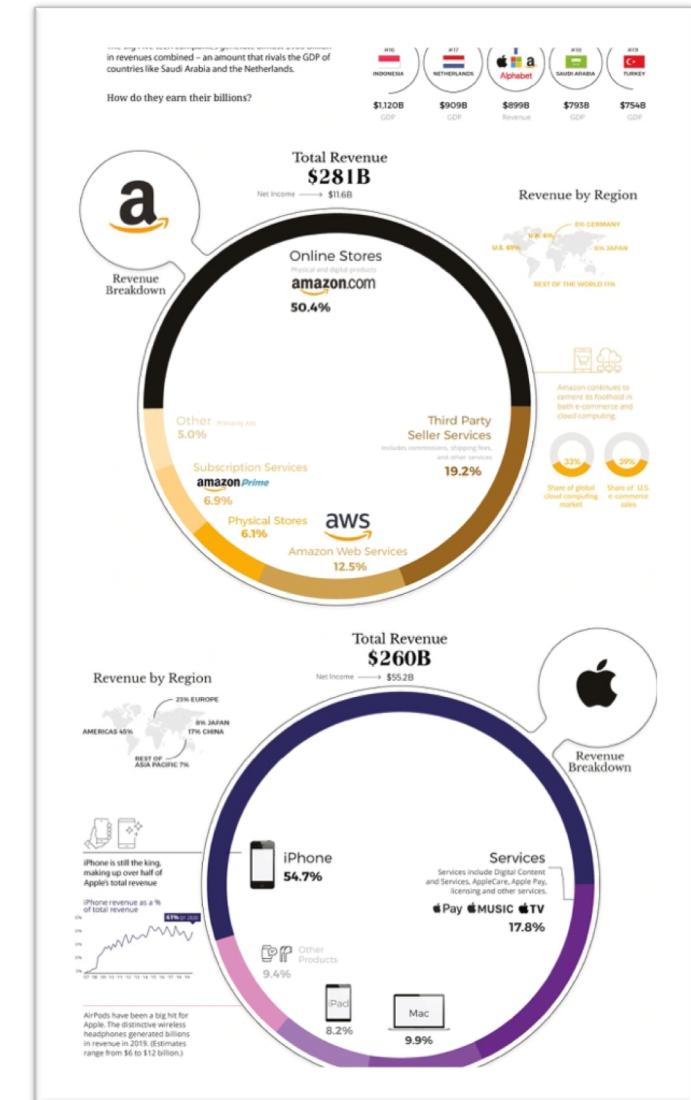
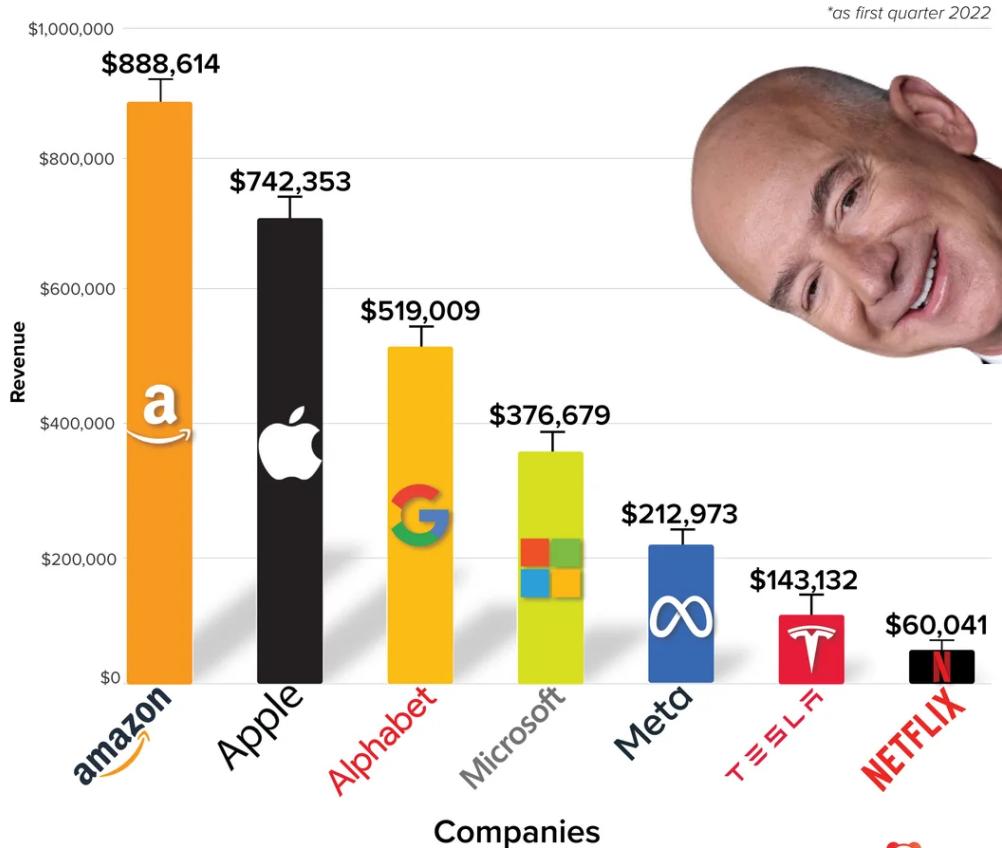
- Expenditure on software represents a **significant** fraction of Gross national product (GNP) GNP in all developed countries.
- Worldwide IT spending is projected to total **\$4.6 trillion** in 2023.



Software Engineering

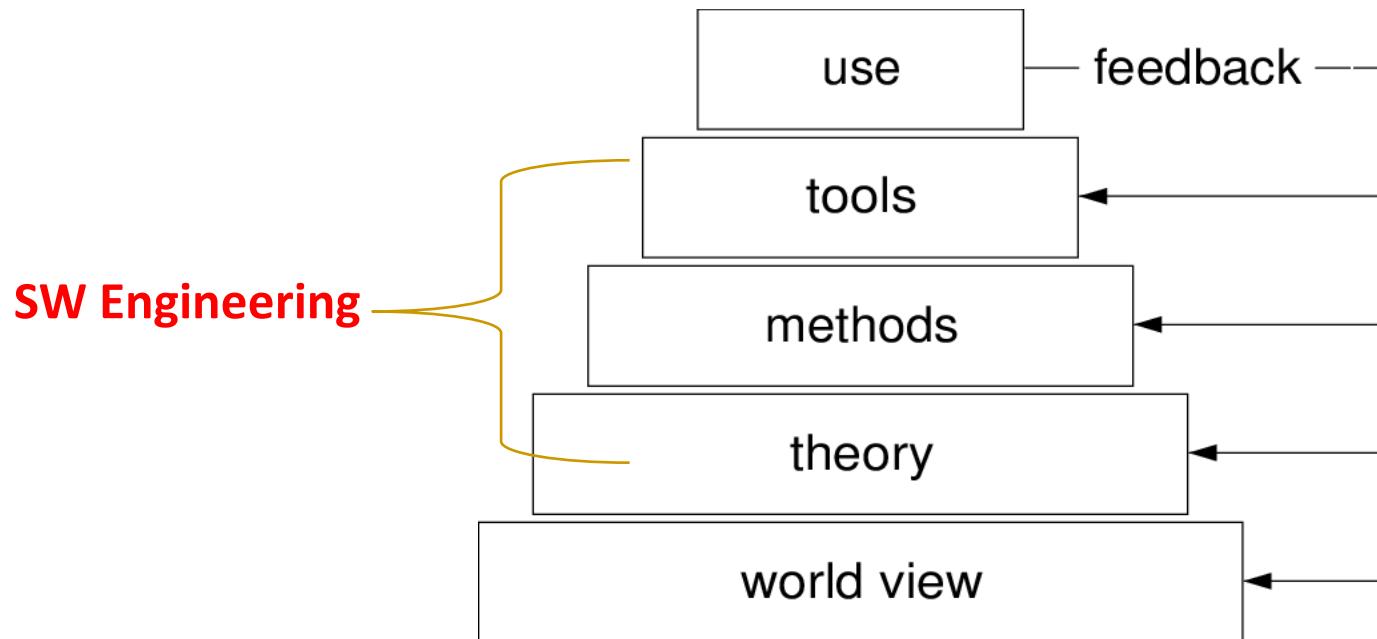
Revenue Per Minute Of Top Tech Companies

(The MATMAAN collective)



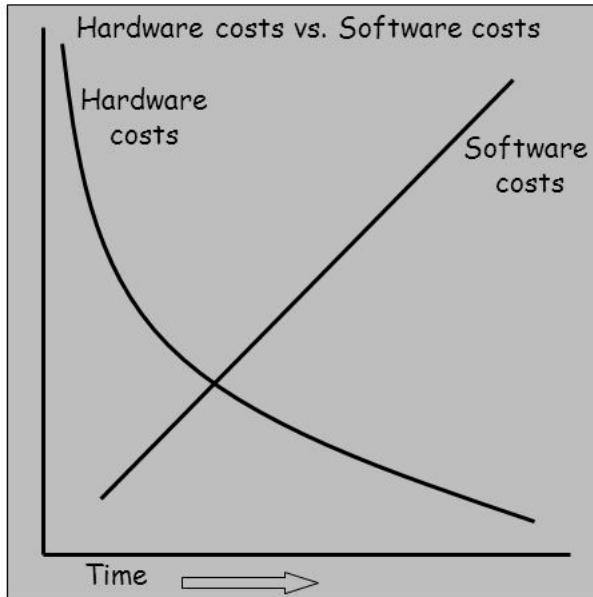
Software Engineering

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



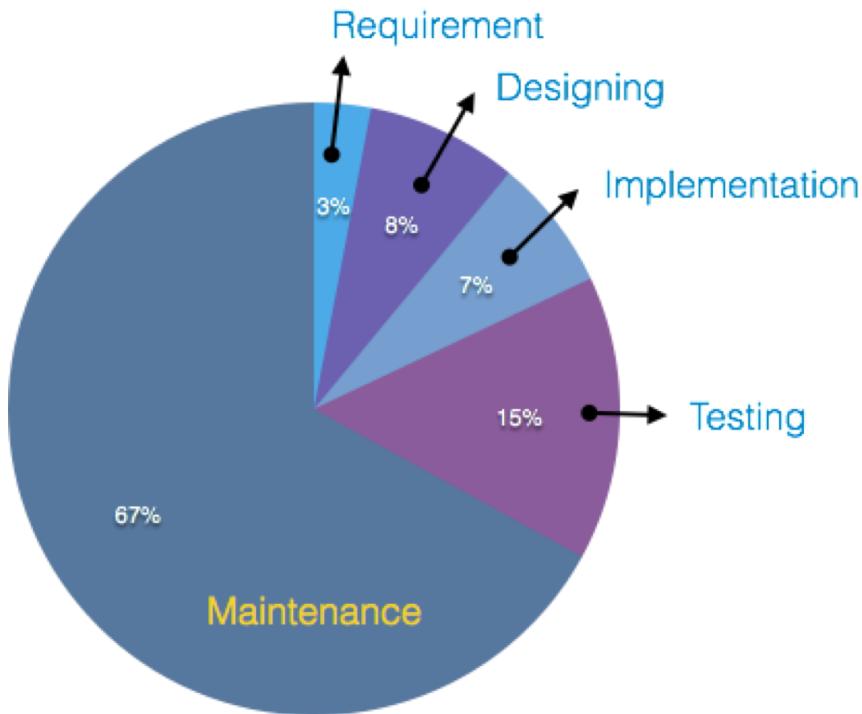
Software Cost

- Software costs often dominate computer system costs. The costs of **software** on a PC are often **greater** than the **hardware** cost.
- Software engineering is concerned with **cost-effective** software development.



Software Cost

- Software costs more to maintain than it does to develop. For systems with a long life, maintenance costs may be several times development costs.
- Software engineering is concerned with **cost-effective** software development.



Software Failure

- There are still many reports of software projects going wrong and of “software failures.”
- Many of these so-called software failures **are a consequence of two factors:**

1- Increasing system complexity

- Systems have to be built and delivered more **quickly; larger**, even more **complex** systems are required; and systems have to have new capabilities that were previously thought to be impossible..

Software Failure

- There are still many reports of software projects going wrong and of “software failures.”
- Many of these so-called software failures **are a consequence of two factors:**

2. Failure to use software engineering methods

- Many development companies do not use software engineering methods in their everyday work. Consequently, their software is often more expensive and less reliable than it should be.

Frequently Asked Questions (FAQs)

1 What is software (SW)?

- Computer **programs** and associated **documentation**.
- Software products may be developed for a **particular customer** or may be developed for a **general market**.



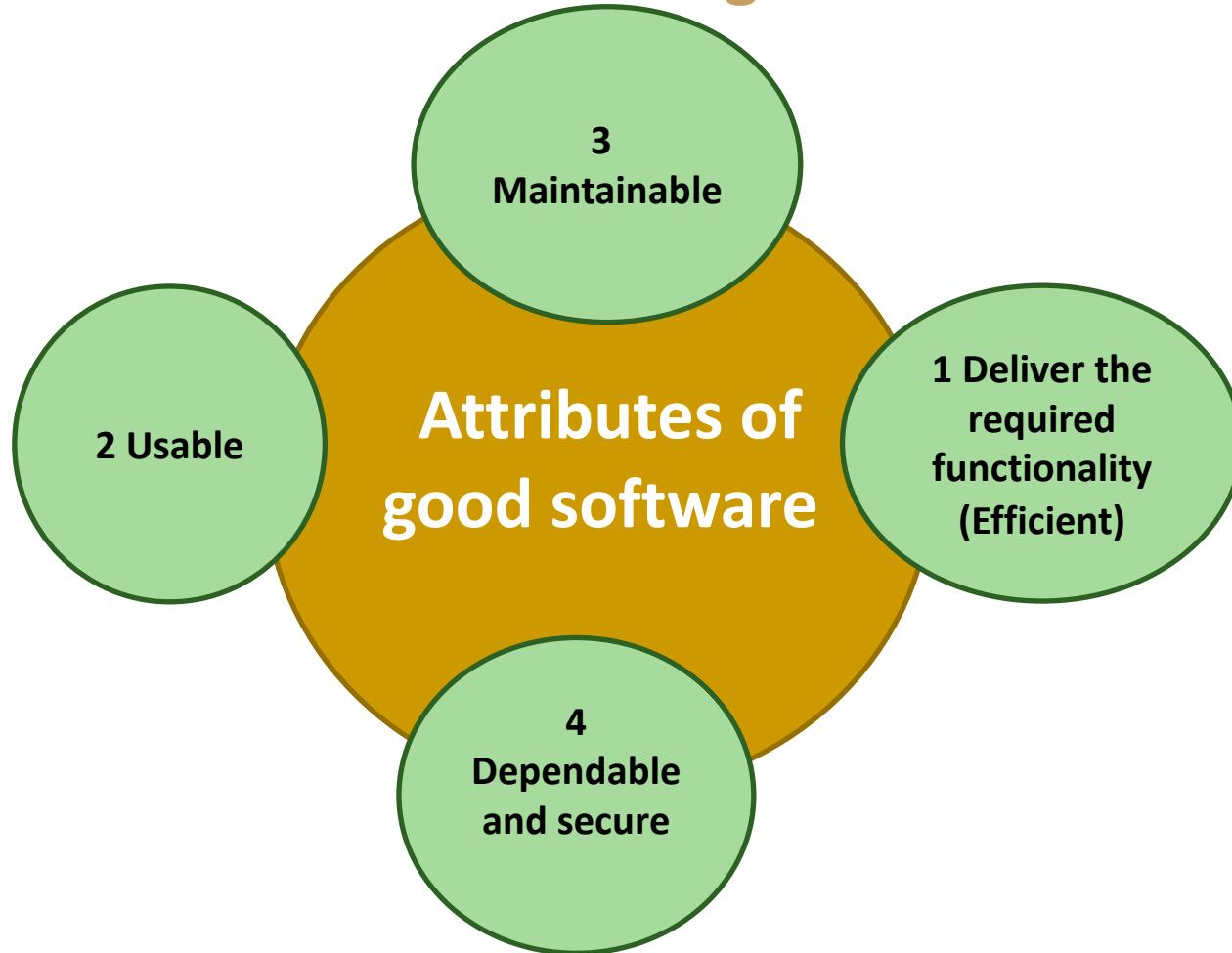
Particular customer



General market

Frequently Asked Questions (FAQs)

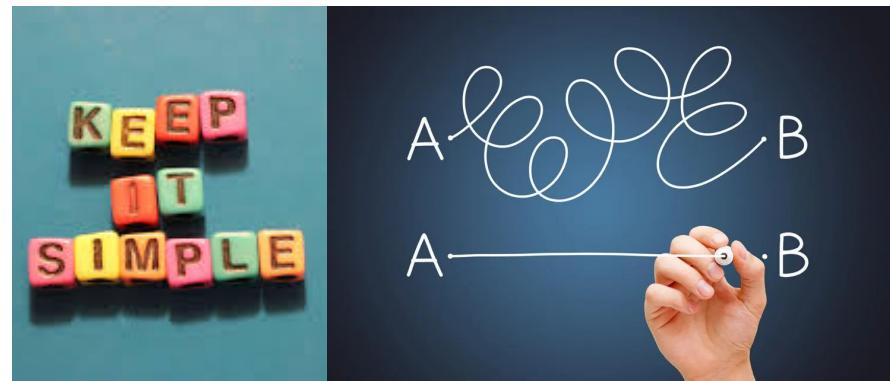
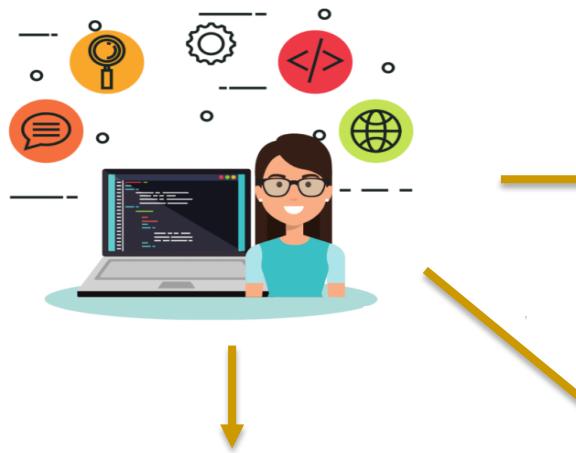
2 What are the attributes of good software?



Frequently Asked Questions (FAQs)

3 What is software engineering?

Applying **Engineering Principles** to all aspects of software production.



Frequently Asked Questions (FAQs)

4 What are the fundamental software engineering activities?

- All software processes include the following **four fundamental software engineering activities**.

Software specification	Defining the software and specifying constraints on its use.
Software development	Designing and building the software.
Software validation	Testing the software to ensure it meets requirements, is efficient, usable, maintainable, dependable and secure .
Software evolution	Modifying the software to meet changes in the business or organisational context.

Frequently Asked Questions (FAQs)

5 What is the difference between software engineering and computer science?



vs

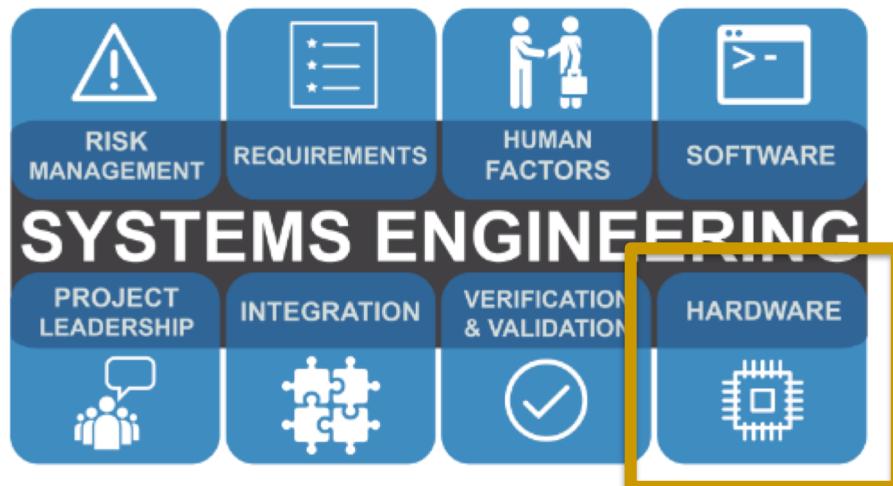


- Theory
- Fundamentals
- Practicalities of :
 - ✓ Developing &
 - ✓ Delivering useful software

Frequently Asked Questions (FAQs)

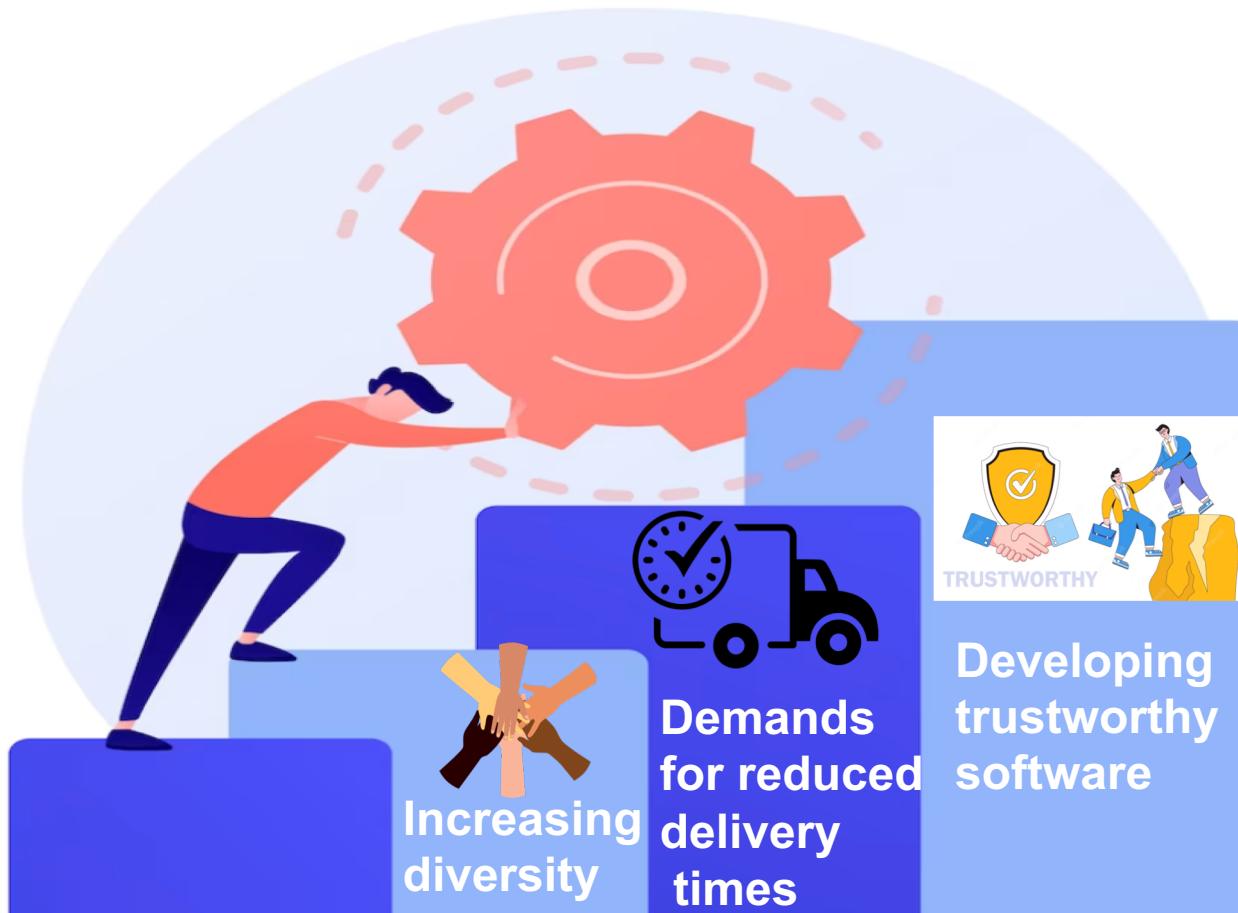
6 What is the difference between software engineering and system engineering?

- **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.



Frequently Asked Questions (FAQs)

7 What are the key challenges facing software engineering?





Frequently Asked Questions (FAQs)

8 What are the costs of software engineering?

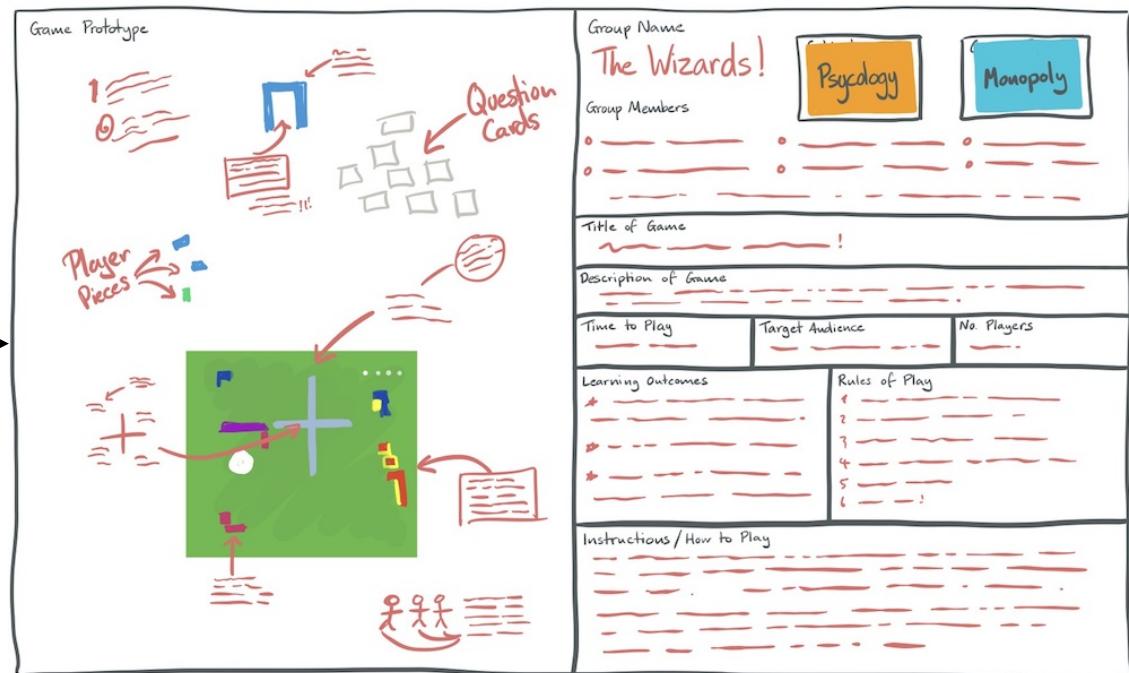
- Roughly 60% of software costs are development costs.
- Second most are the testing costs (40%).
- For custom software, **evolution costs** often exceed development costs.

Frequently Asked Questions (FAQs)

9 What are the best software engineering techniques and methods?

- Different techniques are appropriate for different types of system.

Games ----->



Prototypes

Frequently Asked Questions (FAQs)

9 What are the best software engineering techniques and methods?

- Different techniques are appropriate for different types of system.

Safety critical control systems



Complete and analyzable specification



Frequently Asked Questions (FAQs)

9 What are the best software engineering techniques and methods?

- You can **NOT** say that one method is better than another.
- No universal set of software techniques that is applicable to all of **different types** of software system.
- The software engineering **methods** and **tools** used depend on the type of application being developed, the requirements of the customer and the background of the development team.



Frequently Asked Questions (FAQs)

10 What differences has the web made to software engineering?

- The availability of **software services**.
- The possibility of developing highly **distributed service-based** systems.

Web-based systems development has led to **important advances** in programming languages and software reuse.

FAQs

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 from initial conception to operation and maintenance.
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.
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. There are no methods and techniques that are good for everything.
What differences has the Internet made to software engineering?	Not only has the Internet led to the development of massive, highly distributed, service-based systems, it has also supported the creation of an "app" industry for mobile devices which has changed the economics of software.

Software Products

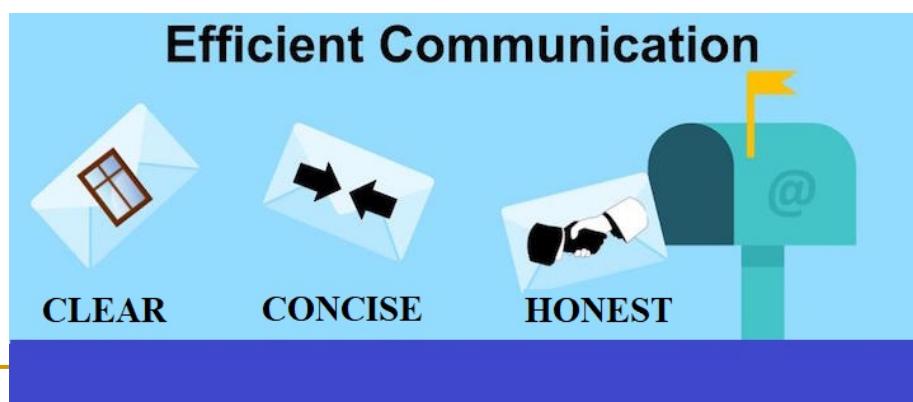
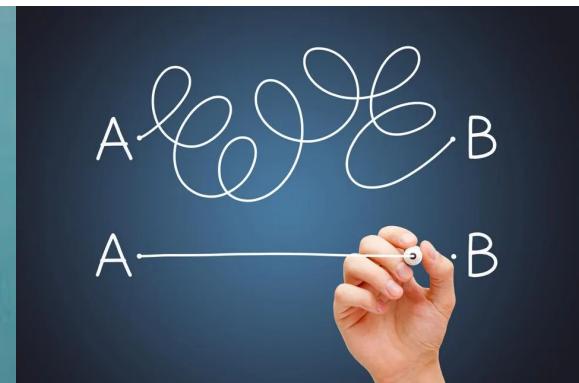
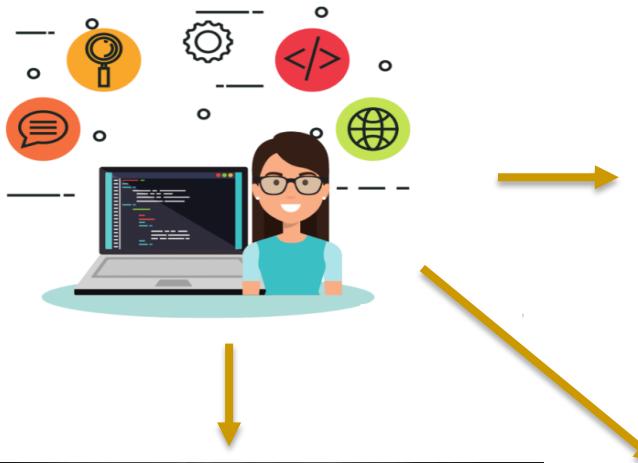
	Generic products	Customized products
Customers	Marketed and sold to any customer who wishes to buy them (stand-alone systems).	a specific customer to meet their own needs (bespoke).
Example	<ul style="list-style-type: none"> • Apps for mobile devices. • Software for PCs (e.g., databases, drawing packages, and project management tools). 	<ul style="list-style-type: none"> • Control systems for electronic devices. • Particular business process. • Air traffic control systems.
Specification and changes	Owned by the software developer	Owned by the customer.

RECALL: FAQs



3 What is software engineering?

Applying **Engineering Principles** to **all aspects of software production.**

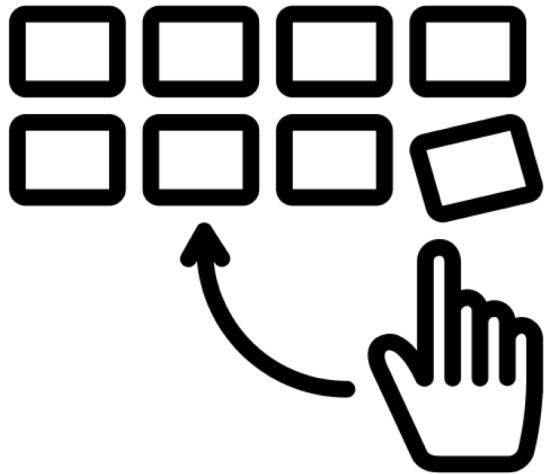


Software Engineering

- **Engineering Principles:**
 - ✓ Engineers **make things work**.
 - ✓ They apply theories, methods, and tools where these are **appropriate**.
 - ✓ Try to discover **solutions** to problems (within organizational and financial **constraints**).
- **All aspects of software production:** Not only **technical processes** of software development, but also **project management** and the **development of tools, methods, and theories**.

Software Engineering

- Adopt a ...



Systematic (process = 4 activities) & **Organized**

Approach



4 What are the fundamental software engineering activities?

- All software processes include the following **four fundamental software engineering activities**.

Software specification	Defining the software and specifying constraints on its use.
Software development	Designing and building the software.
Software validation	Testing the software to ensure it meets requirements, is efficient, usable, maintainable, dependable and secure .
Software evolution	Modifying the software to meet changes in the business or organisational context.

Importance of Software Engineering

1. Individuals and society rely on advanced software systems. We need to be able to produce **reliable** and **trustworthy** systems economically and quickly.

2. **Cheaper**, in the long run, to use software engineering methods. Failure to use software engineering method leads to
 - **higher costs for testing**,
 - **quality assurance**, and
 - **long-term maintenance**.

RECALL: Software Failure



- There are still many reports of software projects going wrong and of “software failures.”
- Many of these so-called software failures **are a consequence of two factors:**
 - 1- Increasing system complexity**
 - 2. Failure to use software engineering methods**

General Issues That Affect SW

1. Heterogeneity

- Increasingly, systems are required to operate as **distributed systems** across **networks** that include different types of computer and mobile devices.



2. 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.

General Issues That Affect SW

3. Security and trust

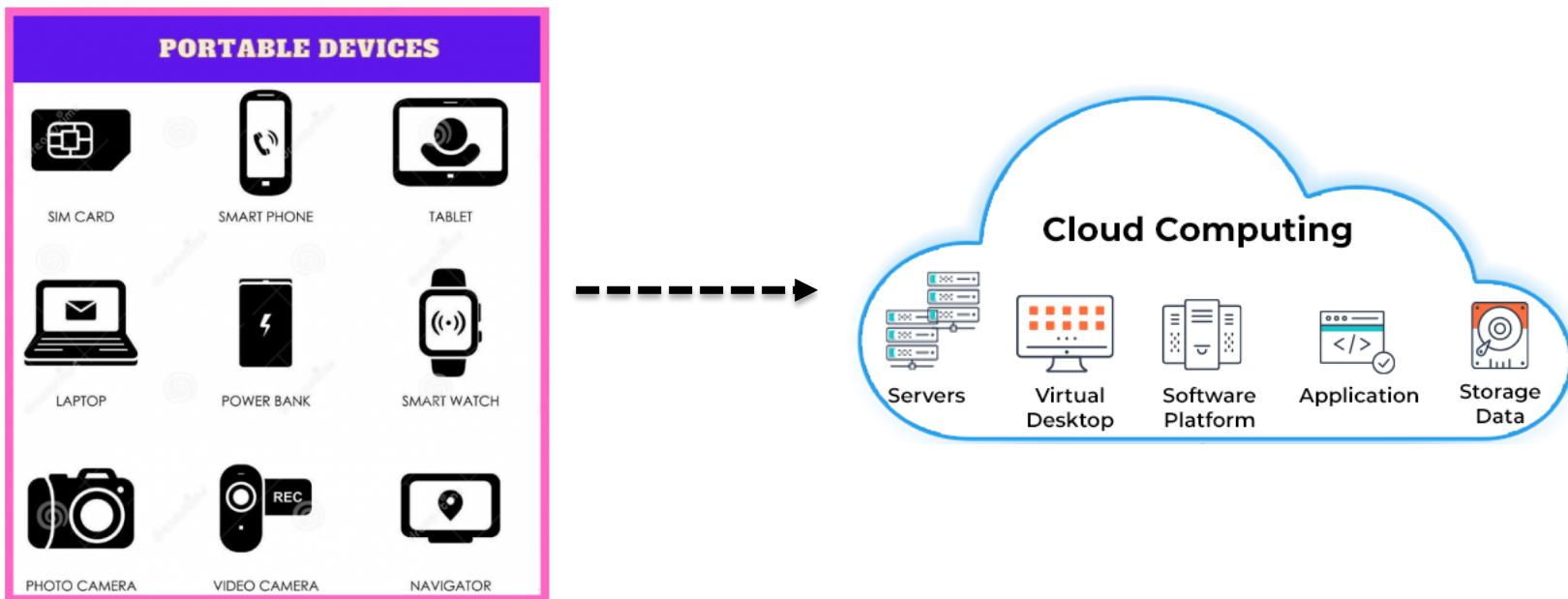
It is essential that we can **trust** that software. This is especially true for **remote software systems** accessed through the web. We have to make sure that malicious users **cannot** successfully **attack** our software .



General Issues That Affect SW

4. Scale

Software has to be developed across a **very wide range of scales**, from very **small** embedded systems in portable or wearable devices through to **Internet-scale**, cloud-based systems that serve a **global** community.





9 What are the best software engineering techniques and methods?

- You can **NOT** say that one method is better than another.
- **Diversity:** No universal set of software techniques that is applicable to all of **different types** of software system.
- The software engineering **methods** and **tools** used depend on the **type of application** being developed, the **requirements** of the **customer** and the **background** of the development team.

Application Types (1/8)

■ Stand-alone applications

- These are application systems that run on a **personal computer** (PC) or **apps run on mobile devices**. They include all necessary functionality (no network connection is needed).
- **Examples:** office applications, and travel apps.



Word



Microsoft Excel



Microsoft Outlook



PowerPoint



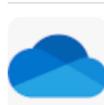
Microsoft OneNote



Office 365



Microsoft Publisher



OneDrive

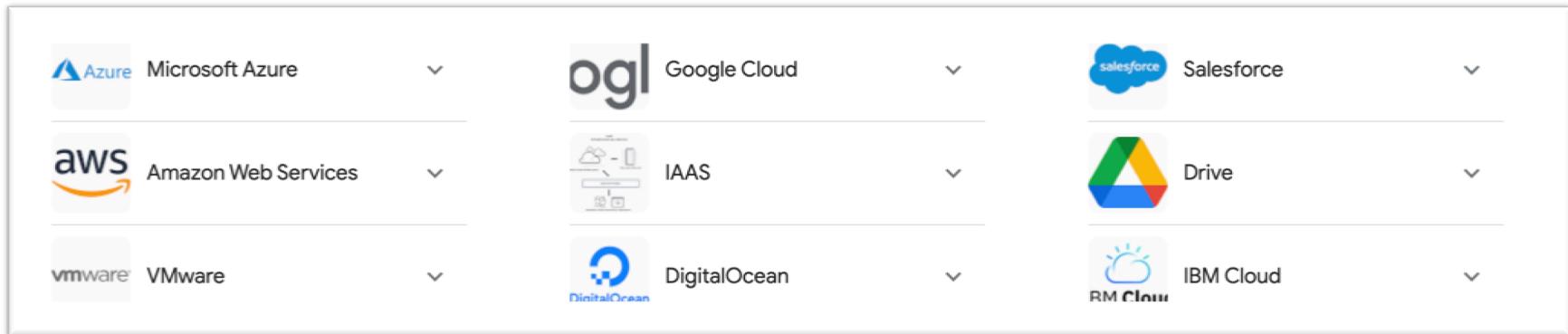


Microsoft Access

Application Types (2/8)

■ Interactive transaction-based applications

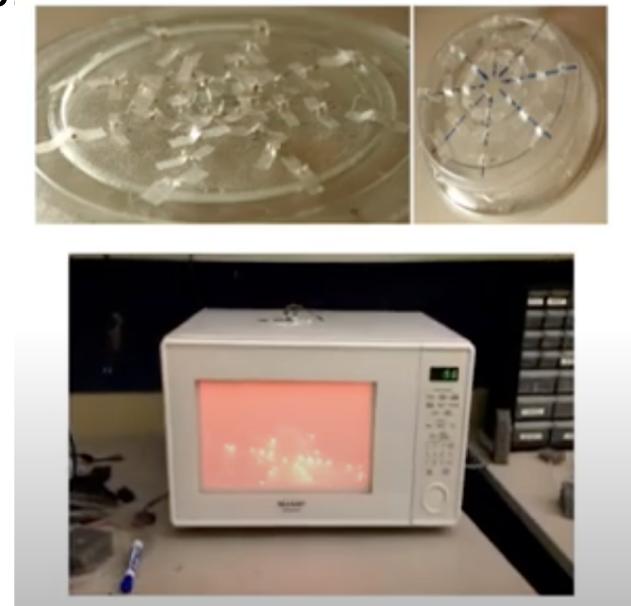
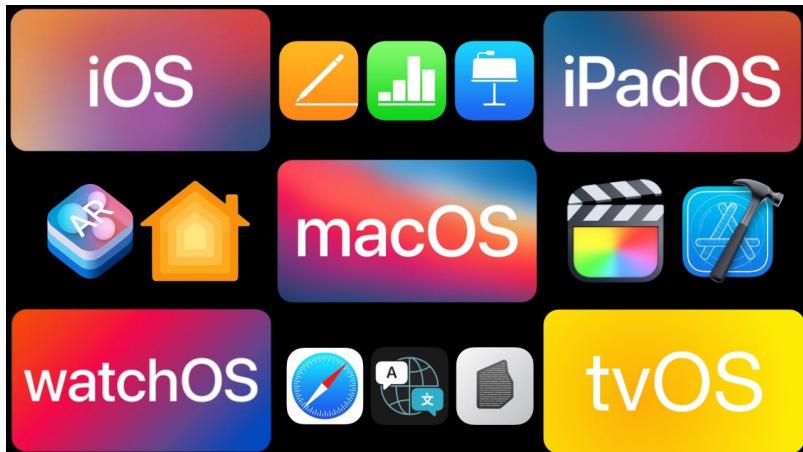
- Apps that execute on a **remote computer** and are accessed by users from their own PCs, phones, or tablets.
- **Examples:** web apps (e-commerce), business system, and cloud-based services.



Application Types (3/8)

■ Embedded control systems (widely used)

- These are software control systems that control and manage hardware devices.
- Examples: software in a phone, and software in a microwave oven to control the cooking process



Application Types (4/8)

■ Batch processing systems

- Designed to process data in large batches. They process large numbers of **individual inputs** to create corresponding **outputs**.
- **Examples:** periodic billing systems, such as phone billing systems, and salary payment systems.



Application Types (5/8)

■ Entertainment systems

- These are systems that are primarily for personal use and which are intended to entertain the user.
- Examples: games.



Application Types (6/8)

■ Systems for modeling and simulation

- These are systems that are developed by scientists and engineers to model physical processes or situations, which include many, separate, interacting objects.
- Examples: MATLAB.



Application Types (7/8)

■ Data collection systems

- System that collect data from their environment using a set of sensors and send that data to other systems for processing.
- Examples: “Big data” analysis (may involve cloud-based systems) carrying out **statistical analysis** and looking for relationships in the collected data.



Application Types (8/8)

■ Systems of systems

- ❑ Used in large organizations, and composed of a number of other **software systems**.

■ Examples: generic products such as Enterprise Resource Planning (ERP) systems.



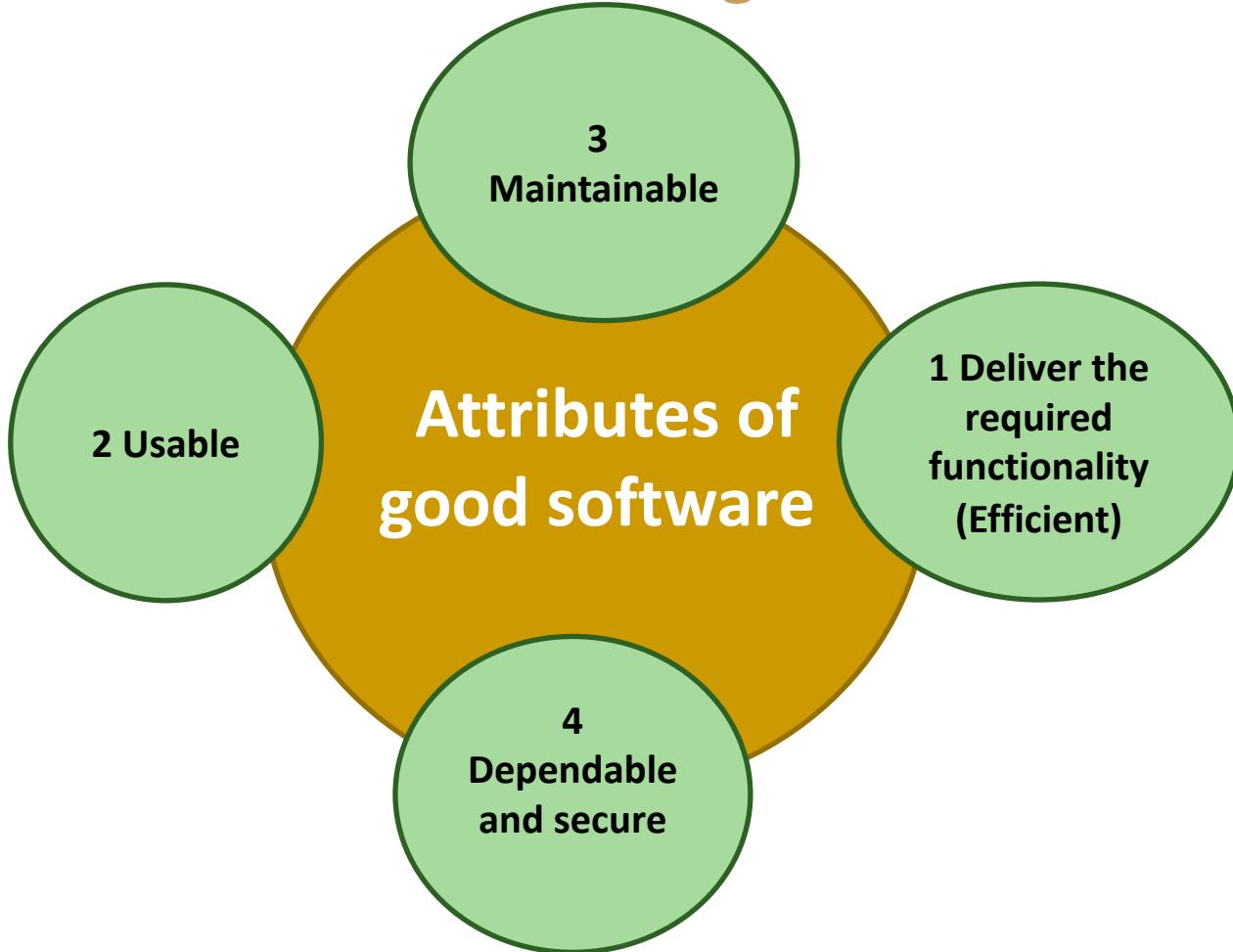


Software Engineering Fundamentals

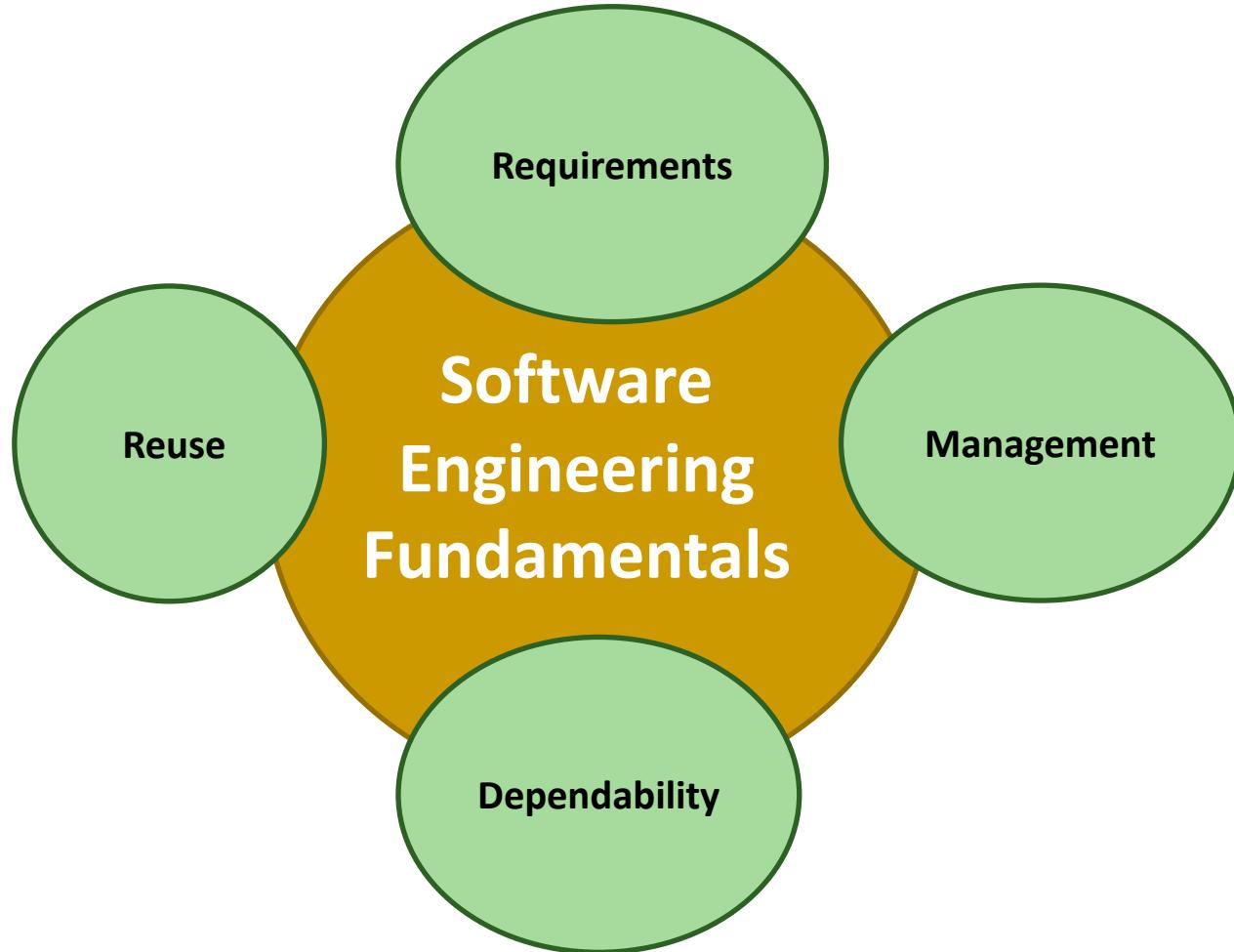
- Some fundamental principles apply to **all types** of software system, irrespective of the development techniques used:
 1. **Management:** Systems should be developed using a managed and understood **development process**.
 2. **Dependability and performance:** SW should **behave** as expected, without failure.
 3. **Requirements:** Understanding and **managing the software specification** and requirements (what the software should do) are important.
 4. **Reuse:** where appropriate, you should **reuse software** that has already been developed rather than write new software.



2 What are the attributes of good software?

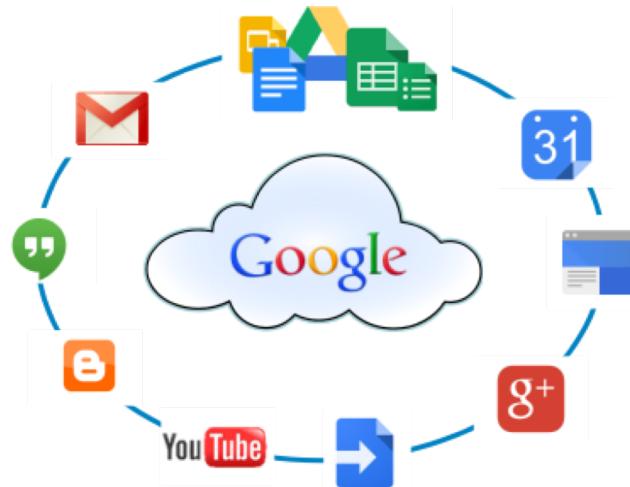


Software Engineering Fundamentals



Internet Software Engineering

- The Web is now a platform for running application and organizations are increasingly developing **web-based** systems rather than **local systems**.
- **Web services** allow application functionality to be accessed over the web.
- **Cloud computing** is an approach to the provision of computer services where applications run remotely on the ‘cloud’.





10 What differences has the web made to software engineering?

- The availability of **software services** (e.g., **Google apps**).
- The possibility of developing highly **distributed service-based** systems.

Web-based systems development has led to **important advances** in programming languages and software reuse.

Web-based Software Engineering

■ Software reuse

- ❑ dominant approach (using pre-existing software components).

■ Incremental and agile development

- ❑ impractical to specify all the requirements in advance.

■ Service-oriented systems

- ❑ stand-alone web services.

■ Rich interfaces

- ❑ AJAX and HTML5.

Do you have
any QUESTION?

