

Class 7: Introduction to Software Engineering

Master Course:

Data-driven Systems Engineering (ML Operations)

440MI and 305SM

Agenda

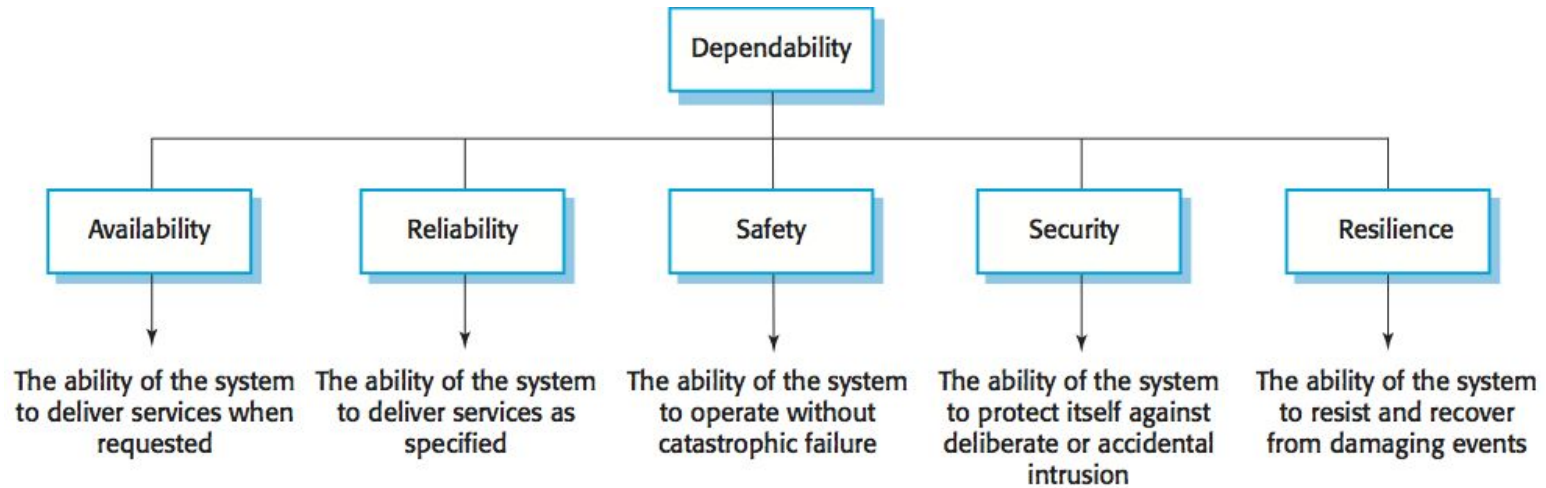
- What is Good Software?
- Professional software process activities
- Software engineering diversity
- Software engineering fundamentals
- Software Engineering
- Software Costs
- Software Products
- General Questions

What is a good software?

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.

Dependability...



Importance of software engineering

- ✧ 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.

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.

Software process activities - **specification**

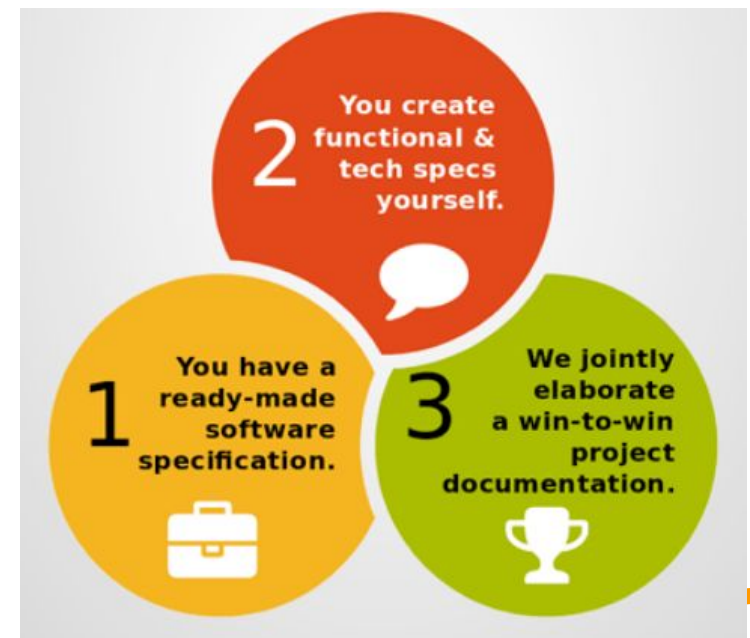
- ✧ Software **specification**, where customers and engineers define the software that is to be produced and the constraints on its operation.

Software specification usually contains



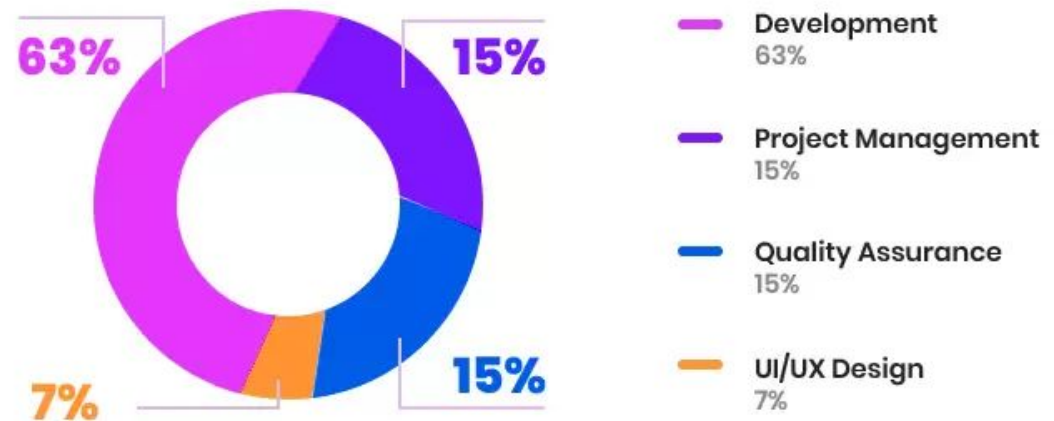
Software process activities - **specification**

- ✧ Software **specification**, where customers and engineers define the software that is to be produced and the constraints on its operation.



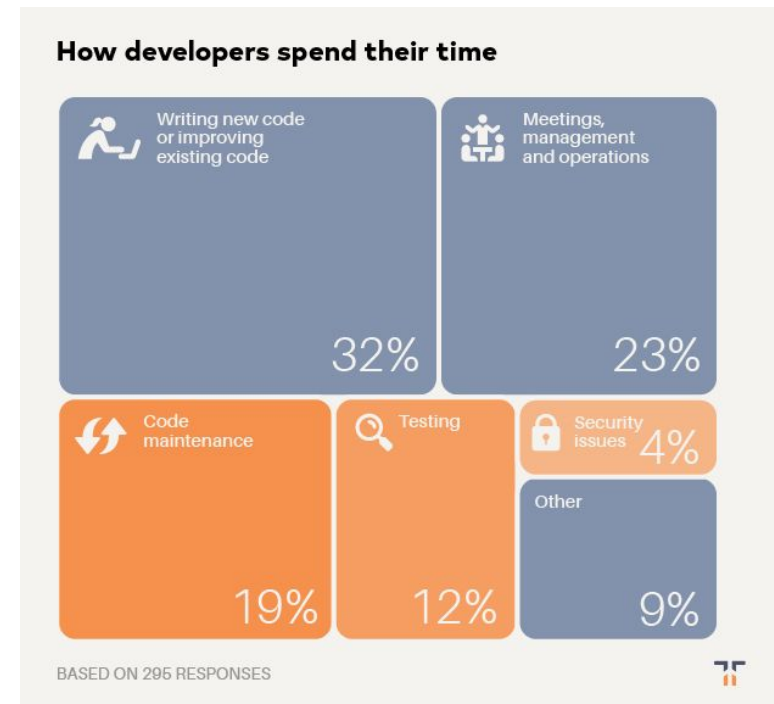
Software process activities - **development**

- ✧ Software **development**, where the software is designed and programmed.



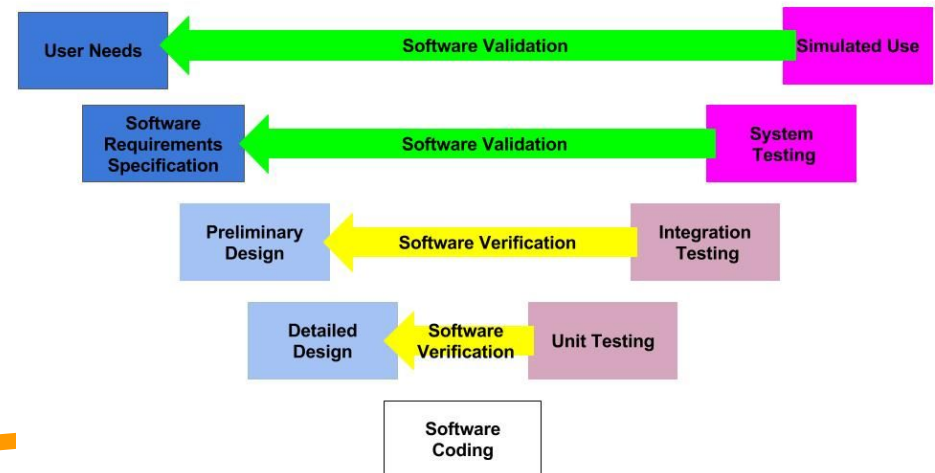
Software process activities - **development**

- ✧ Software **development**, where the software is designed and programmed.



Software process activities - **validation**

- ✧ Software **validation**, where the software is checked to ensure that it is what the customer requires.
- ✧ Software **verification** regards specific requirements in the scope of software and technology.



Software process activities - **evolution**

- ✧ 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.

General issues that affect most software

Business and social change

The Internet of Things

● Connected devices ● World population



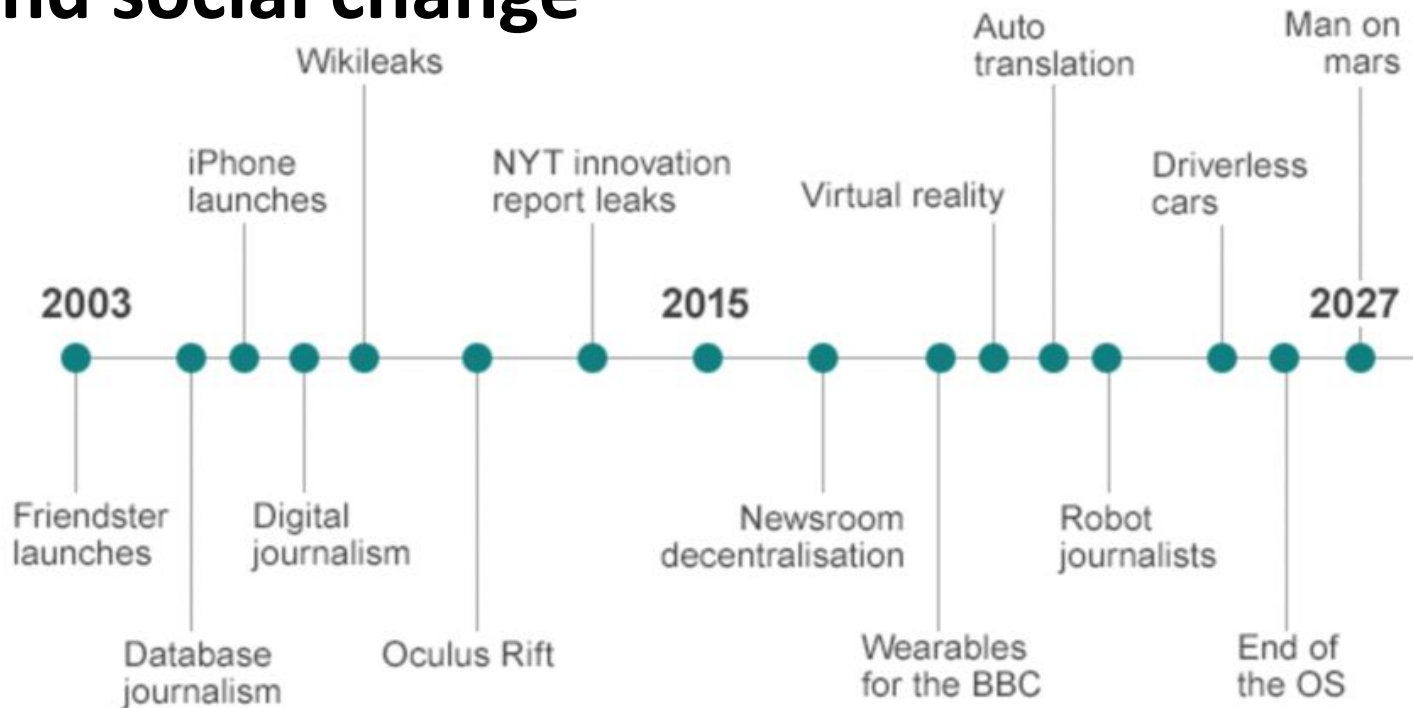
* Separate HP forecast based on different methodology

Source: Cisco IBSG, Hewlett-Packard

BBC

General issues that affect most software

Business and social change



Software engineering diversity

- ✧ There are many different types of software system and there is **no universal set of software techniques** that is applicable to all of these.
- ✧ 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

Stand-alone applications

- These are application systems that **run on a local computer**, such as a PC. They include all necessary functionality and do not need to be connected to a network.

Interactive transaction-based applications

- Applications that execute on a **remote computer** and are accessed by users from their own PCs or terminals. These include web applications such as e-commerce applications.

Embedded control systems

- These are **software control** systems that control and manage **hardware** devices. Numerically, there are probably more embedded systems than any other type of system.

Application types

Batch processing systems

- These are business systems that are designed to process data in **large batches**. They process large numbers of individual inputs to create corresponding outputs.

Entertainment systems

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

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.

Application types

Data collection systems

- These are systems that collect data from their environment **using a set of sensors** and send that data to other systems for processing.

Systems of systems

- These are systems that are composed of a number of other software systems.

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**.

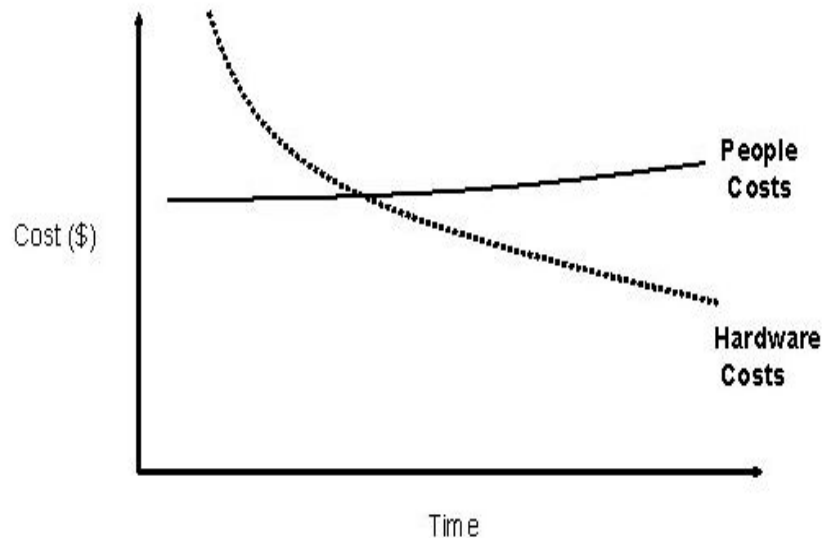
- It is now generally recognized that it is impractical to specify all the requirements for such systems in advance.

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

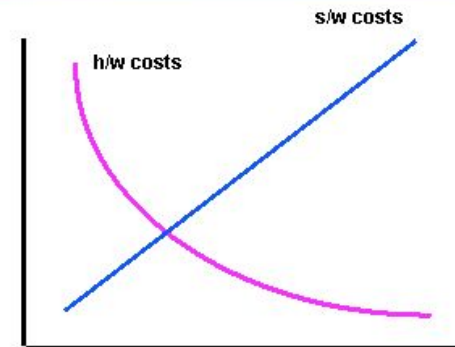
- Technologies such as AJAX allow rich interfaces to be created within a web browser but are still difficult to use. Web forms with local scripting are more commonly used.

Software costs

- Software costs often dominate **computer system costs**. The costs of software on a PC are often greater than the hardware cost.



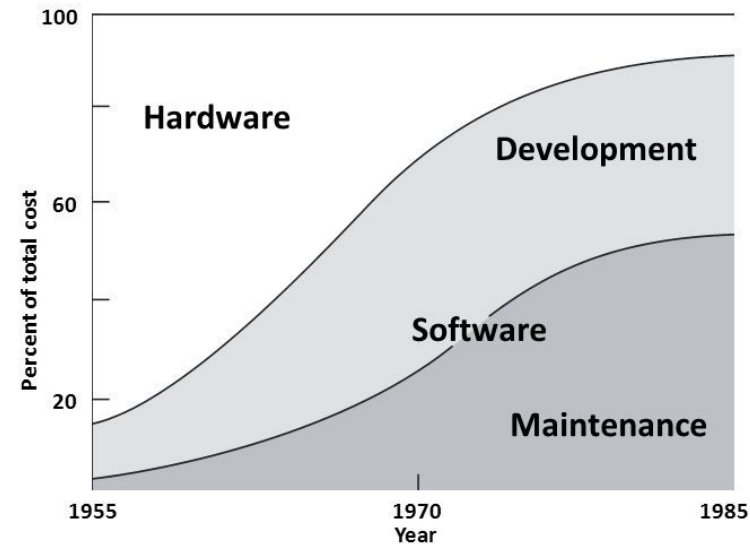
Hardware Costs vs Software Costs
(% of overall costs)



Software costs

- 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.

Relative distribution of software/hardware costs



SE, Maintenance, Hans van Vliet, ©2008

Software costs

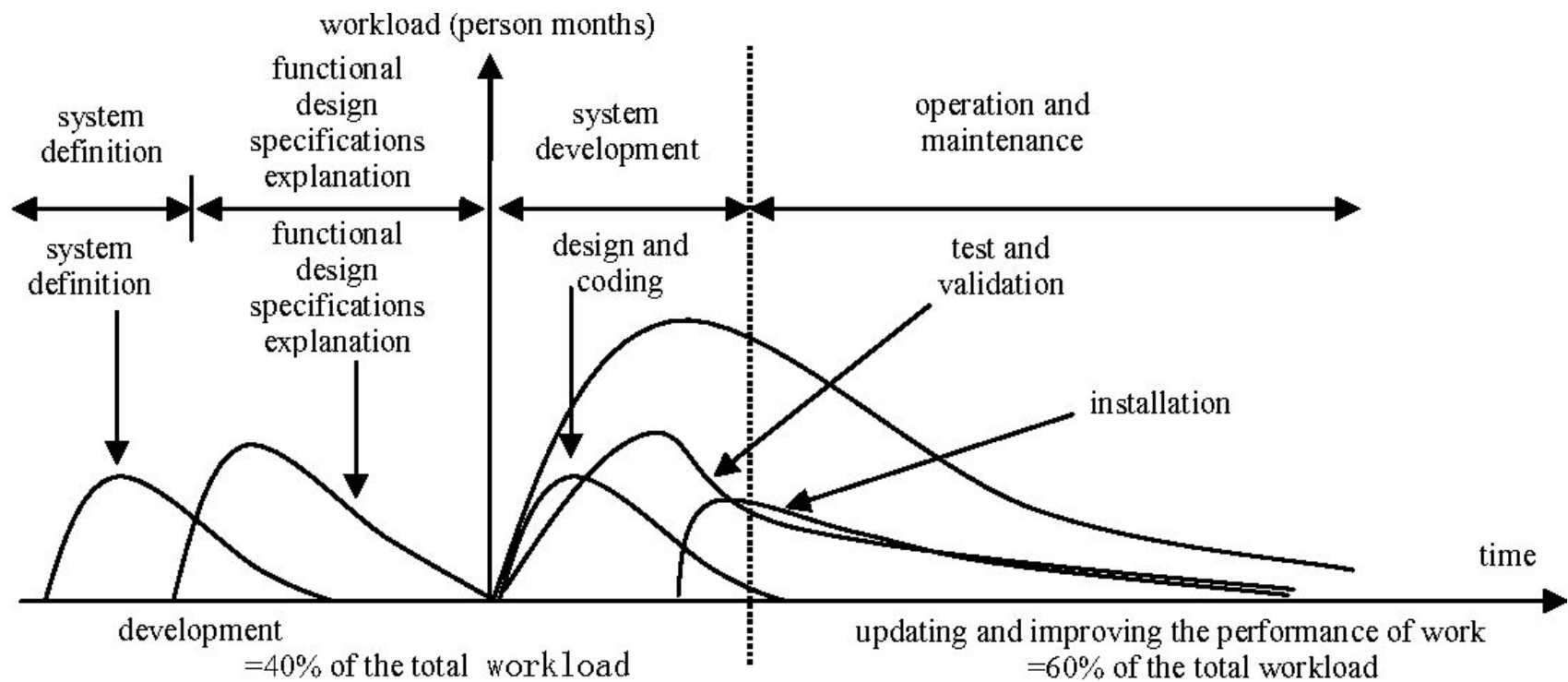


Figure 1. The workload of large-scale distribution of software projects

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.

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.

Key Insights on MLOps

ML Operations)
MI and 305SM



Textbook

