Introduction To Software Engineering

Presented by:

Basma Hussien Mohamed

Evaluation Criteria

- 30% Class Participation & Assignments
- 70% Presentation

- Introduction
- Professional software development
- Software engineering ethics
- Case studies

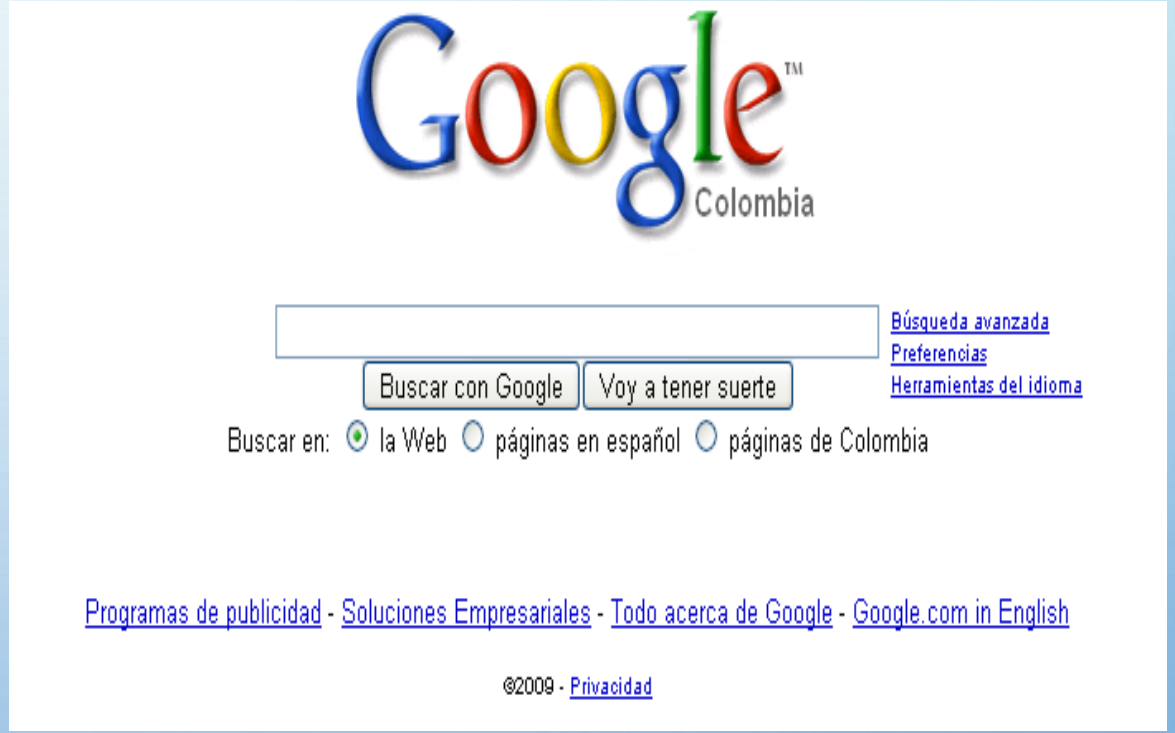
Importance Of Software

Software Can Have A Huge Impact

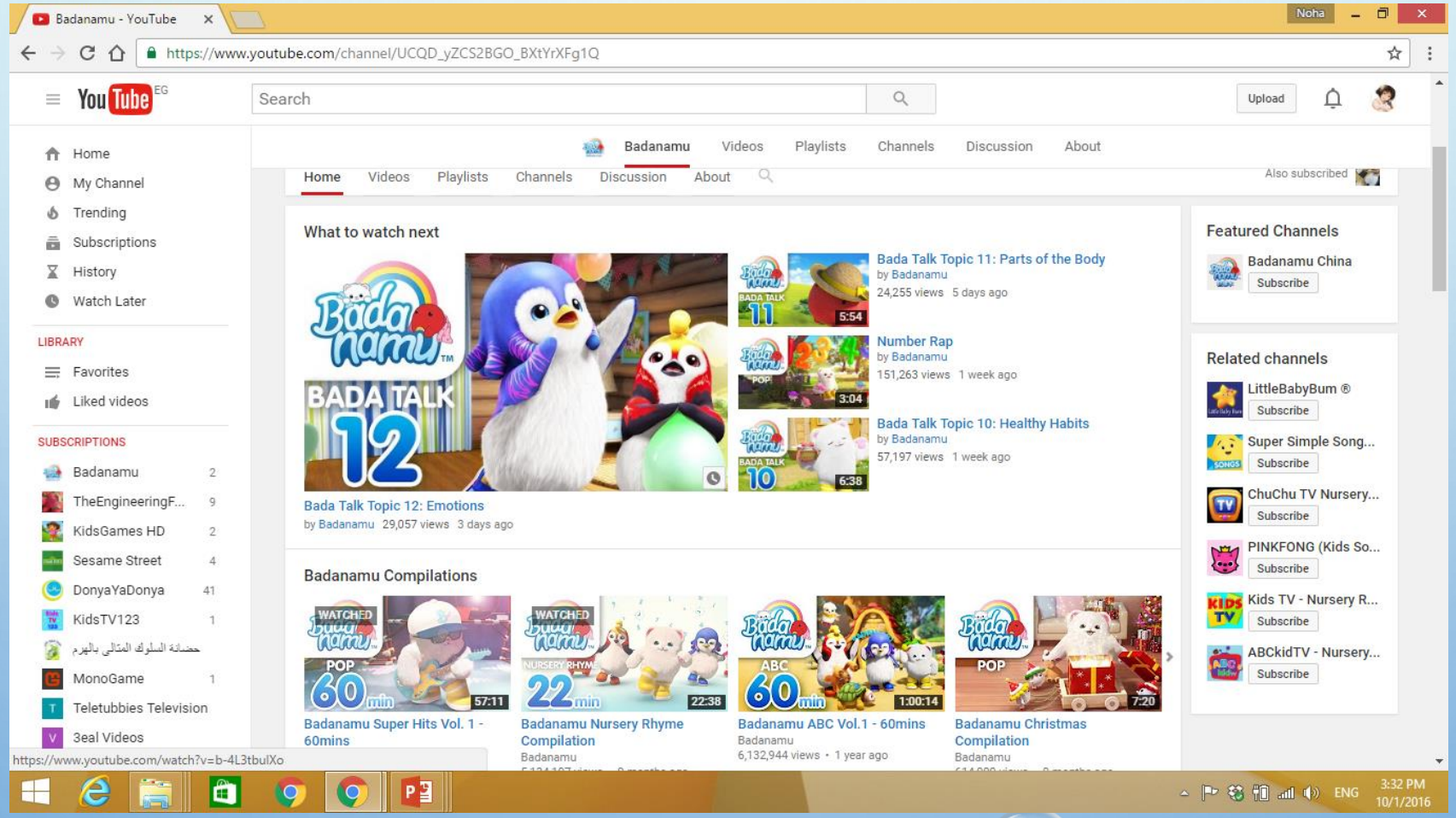
In Any Aspect Of Society.

Where can you find Software ?

Some popular ones…



Some popular ones…



And even in…


Sony Ericsson
WALKMAN
UNOFFICIAL CLUB SONY ERICSSON


AT&T
9:42 AM
Text
Calendar
Photos
Camera
YouTube
Stocks
Maps
Weather
Clock
Calculator
Notes
Settings
iTunes
App Store
Phone
Mail
Safari
iPod

Software is Almost Everywhere.

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, as well as methods and tools for professional software development.

Software 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. For systems with a long life, maintenance costs may be several times development costs.
- Software engineering is concerned with cost-effective software development.

Ariane 5 Flight 501

Cause: design errors in the software

Conclusion

Programming is NOT enough!

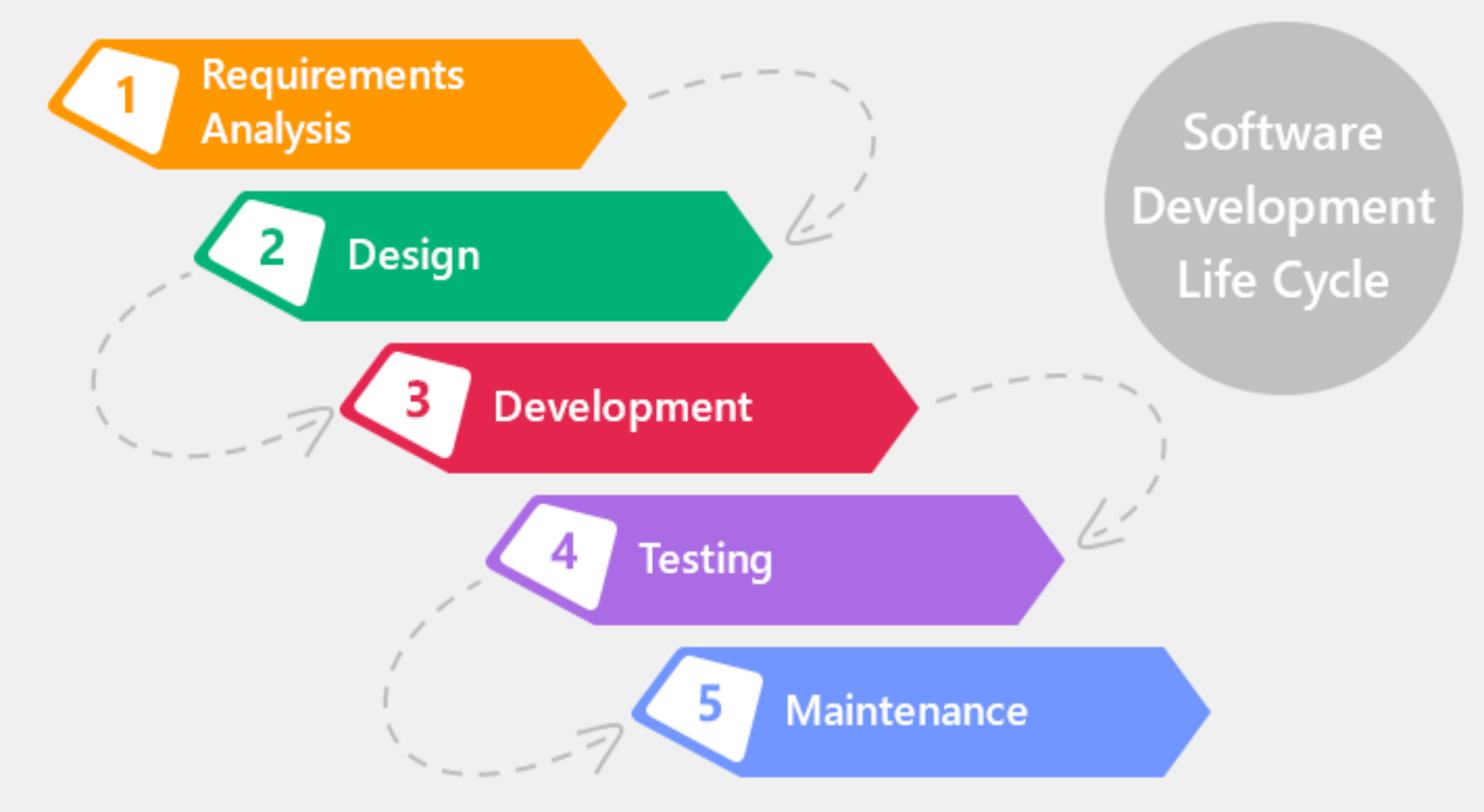
It is not enough to do your best: you must
Know what to do, and THEN do your best.
-- *W. Edwards Deming*

And Since…

A clever person solves a problem.
A wise person avoids it.
- *Albert Einstein*

Professional software development

SDLC


1
Requirements Analysis
2
Design
3
Development
4
Testing
5
Maintenance
Software Development Life Cycle

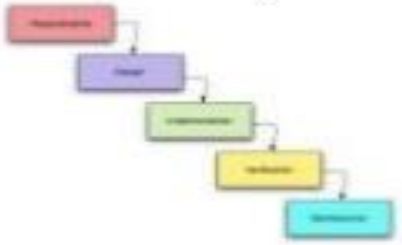
Software Development Process Models

Software Development

Waterfall

70s, 80s

Sequential
Process
All design front-up
Process heavy



RAD

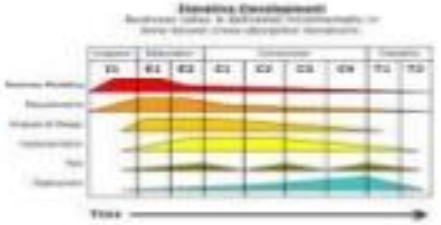
Rapid Application Development
80s, 90s

Rapid Prototyping
Prototype not plan
Process Light

RUP

Rational Unified Process
90s, 00s

Framework for
iterative development
Can be process heavy



Agile

00s, 10s

Iterative and
incremental
Can be process light

Frequently Asked Question About Software Engineering

What is software?

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?

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?

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?

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?

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?

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

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.

What are the key challenges facing software engineering?

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?

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?

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?

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.

Frequently asked questions about software engineering

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.

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 specification
  - 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 specification
  - 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 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.

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.

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 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.

General issues that affect software (Cont.)

- Security and trust
  - As software is intertwined with all aspects of our lives, it is essential that we can trust that software.
- 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.

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 (Cont.)

- 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 (Cont.)

- 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.**

Software Engineering And WEB

- 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’.
  - Users do not buy software buy pay according to use.

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 apply to web-based software in the same way that they apply to other types of software system.

Web-based software engineering

- Software reuse
  - 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.
- Incremental and agile development
  - 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.

Web-based software engineering

- Service-oriented systems
  - Software may be implemented using service-oriented software engineering, where the software components are stand-alone web services.
- Rich interfaces
  - User interfaces are constrained by the capabilities of web browsers.
  - Technologies such as AJAX and Bootstrape 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.

Lets Sum it up…

- 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.

- 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.

- Plan Driven Development : Water Fall Model
- Incremental Developement or RAD:
  - Incremental Model
  - ProtoType Model

  - Reusabilty Development: Reuse oriented Model
  - Agile Development : Extreme programming model

Software engineering Ethics

Software engineering ethics

- Software engineering involves wider responsibilities than simply the application of technical skills.

- Software engineers must behave in an honest and ethically responsible way if they are to be respected as professionals.

- Ethical behaviour is more than simply upholding the law but involves following a set of principles that are morally correct.

Issues of professional responsibility

- **Confidentiality**
  - Engineers should normally respect the confidentiality of their employers or clients irrespective of whether or not a formal confidentiality agreement has been signed.
- **Competence**
  - Engineers should not misrepresent their level of competence. They should not knowingly accept work which is out with their competence.

Issues of professional responsibility

- **Intellectual property rights**
  - Engineers should be aware of local laws governing the use of intellectual property such as patents, copyright, etc. They should be careful to ensure that the intellectual property of employers and clients is protected.
- **Computer misuse**
  - Software engineers should not use their technical skills to misuse other people's computers. Computer misuse ranges from relatively trivial (game playing on an employer's machine, say) to extremely serious (dissemination of viruses).

ACM/IEEE Code of Ethics

- The professional societies in the US have cooperated to produce a code of ethical practice.
- Members of these organisations sign up to the code of practice when they join.
- The Code contains eight Principles related to the behaviour of and decisions made by professional software engineers, including practitioners, educators, managers, supervisors and policy makers, as well as trainees and students of the profession.

Ethical principles

1. **PUBLIC** - Software engineers shall act consistently with the public interest.
2. **CLIENT AND EMPLOYER** - Software engineers shall act in a manner that is in the best interests of their client and employer consistent with the public interest.
3. **PRODUCT** - Software engineers shall ensure that their products and related modifications meet the highest professional standards possible.
4. **JUDGMENT** - Software engineers shall maintain integrity and independence in their professional judgment.
5. **MANAGEMENT** - Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.
6. **PROFESSION** - Software engineers shall advance the integrity and reputation of the profession consistent with the public interest.
7. **COLLEAGUES** - Software engineers shall be fair to and supportive of their colleagues.
8. **SELF** - Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession.

Ethical dilemmas

- Your employer acts in an unethical way and releases a safety-critical system without finishing the testing of the system.
- Participation in the development of military weapons systems or nuclear systems.
- Disagreement in principle with the policies of senior management.

Case studies

Case studies

- A personal insulin pump
  - An embedded system in an insulin pump used by diabetics to maintain blood glucose control.
- A mental health case patient management system
  - Mentcare. A system used to maintain records of people receiving care for mental health problems.
- A wilderness weather station
  - A data collection system that collects data about weather conditions in remote areas.

Key points

- Software engineers have responsibilities to the engineering profession and society. They should not simply be concerned with technical issues.

- Professional societies publish codes of conduct which set out the standards of behaviour expected of their members.

Thank you ☺