Chapter 1:

- Software Crisis: inability of software companies to deliver high quality products on schedule. Projects also became over budget.

- Why SE?
  - Complexity reduced
  - Flexibility increased
  - Can accommodate change
  - Time to develop reduce
  - Cost Effective

- What is SE?
  - Engineering discipline concerned with all aspects of software production from system specification through to deployment and maintenance.
  - Also includes project management tools and devops in order to support software production.

- Maintenance is more costly as compared to development.

- Why Projects fails?
  - Increasing system complexity, however SE techniques helps us build such complex projects in short span of time
  - Failure to use SE methods, causes unreliable and unmaintainable code to be written

- Software: Programs along with relevant documentation. Maybe for a single user to general market

- Attributes of a good software:
  - Maintainable:
    - should be easier / adapting to change with customer requirements
  - Flexible
  - Security:
    - Should be secure: no unwanted access of data
    - Should be safe: not life threatening, or loss of capital
  - Efficient:
    - should not waste system resources, memory cycles. Responsive and good memory utilization
  - Should deliver required functionality to user
    - also easier for other users along with compatibility with other systems

- Fundamental SE activities:
    - Software Specification: (Requirements Specification) *Critical Phase
        - customers define what is to be produced and engineers decide on the constraints of its development
    - Software Development:
        - designing and programming
    - Software Validation:
        - testing for errors and whether output satisfies user requirements
    - Software Evolution:
        - modified to meet changing user and market demands

- SE v/s CS:
    - CS: focuses on theory / fundamentals
    - SE: focuses on practicality of development and delivery of software

- Key Challenges in SE:
    - Increased diversity
    - Developing trustworthy software
    - Reduced delivery times

- Software Products:
    - Generic:
        - general purpose / off the shelf software for catering needs of general market. Such as MS Office etc.
        - Specification and decision on software changes rest with the developer
    - Customized
        - Bespoke / Tailor made software, made to meet requirements of a specific customer. Example embedded systems, traffic control system
        - Customer decides pertaining to specification and changes

- General Issue affecting software:
    - Heterogeneity:
        - need for distributable software to be run over multiple devices mobile, computing etc.
    - Business and social change:
        - ability to change existing software and to develop new software based on new tech
    - Security and trust:
        - should be secure and safe

    - Scale:
        - should be developed for a wide variety of systems from small embedded systems to cloud based system

- SE Diversity:
    - SE methods depend on type of application, platform and customer requirements.
    - Application types:
        - Stand-alone:
            - application on a personal desktop, without network connectivity
        - Transaction-based:
            - applications on remote servers accessed by users from own terminals such as e-commerce applications
        - Embedded:
            - that control, manage hardware devices
        - Batch processing:
            - designed to process large batches of data
        - Entertainment:
            - for personal use, entertainment e.g consoles
        - Modelling / Simulation:
            - to model / simulate scientific process
        - Data-Collection:
            - for recording data using sensors and sending to other systems
        - Systems of systems:
            - Composed of a number of software as SAP ERP systems (avoids data duplications and centralized)

- SE Fundamentals:
    - should be developed using a managed and understood developed process
    - should have good software attributes as dependability, efficiency etc.
    - should display clear understanding of user and software specification
    - should be reused where appropriate instead of writing again

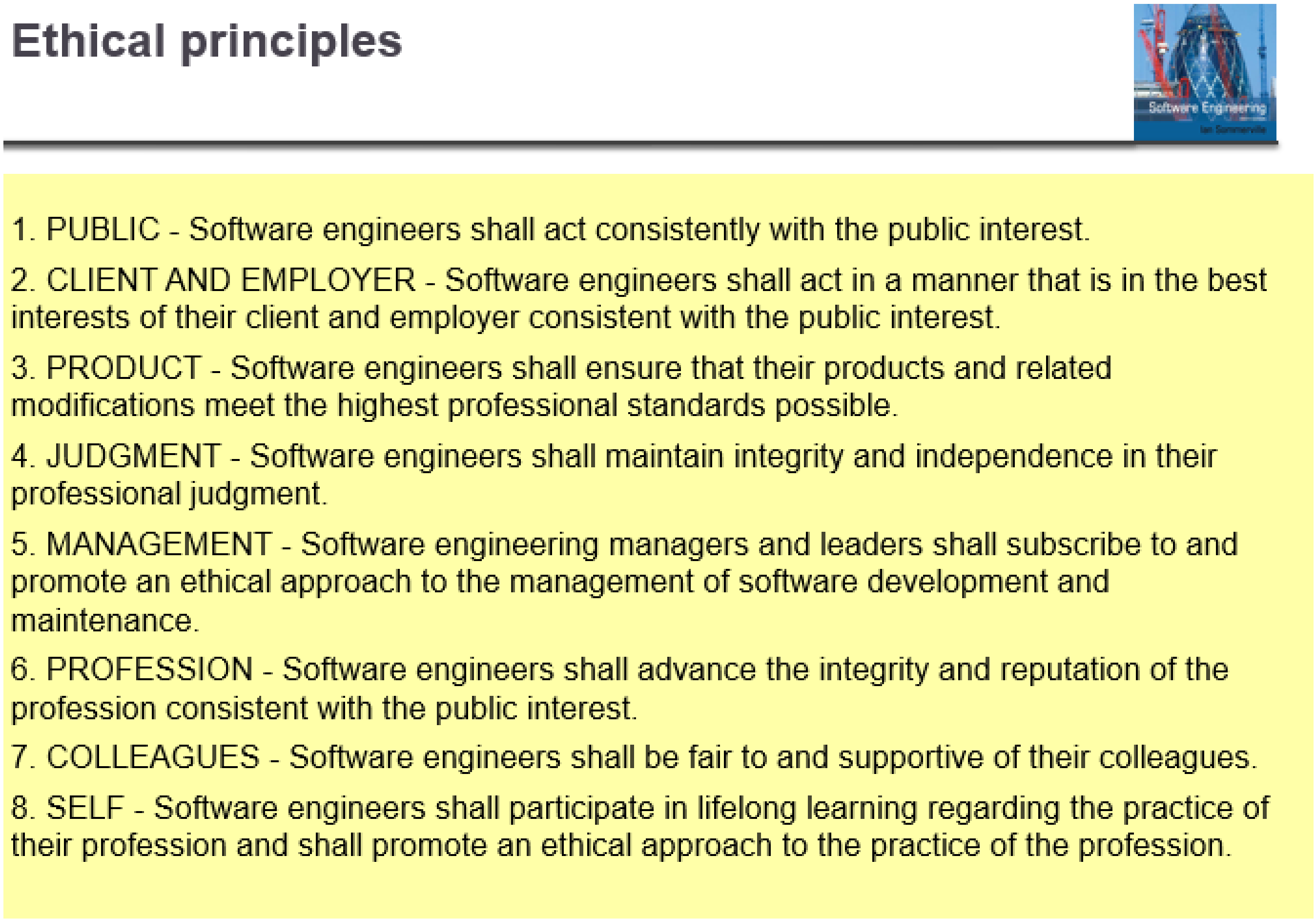
- Web based SE:
    - replacing need for local systems
    - allow functionality to be accessed over web, hence can be used on any platform
    - Cloud computing: applications can now be accessed remotely without having to install on pc, payed on use
    - Advantages:

- Software reuse
- Incremental and agile development:
    - no need to specify its requirements in advance. Can be delivered in ‘increments’
- Service-oriented systems
- Rich Interfaces (user friendly GUI)

- SE Ethics:
    - Confidentiality:
        - should respect privacy of employers or clients, irrespective of agreement
    - Competence:
        - should not accept work in which they have a lower expertise
    - Intellectual property rights:
        - intellectual property of clients, employers should be protected.
    - Computer misuse:
        - should not use technical skills to misuse other’s computer

- Case study:
    - Embedded systems
        - should be safety-critical
        - perform reliably, deliver when required
    - Management systems
        - centralized systems
        - does not require much network connectivity (can be downloaded when online)
        - support CRUD operations
        - Report generations
        - Privacy (security) ensured. Also safety(provide data when needed)
    - Data collection systems
        - allow for maintenance / dynamic reconfiguration in case of system failure
        - efficient data processing and data transmission
        - Suitable data archiving
        - Good power management
    - Digital learning environment (Service oriented)
        - independently accessed
        - rich user interface
        - provide configuration services, application services

READ THIS IF TOO MUCH FREE!



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.