

Software Engineering, Assignment #05

20B030299
Rymbayeva Anelya
FIT, 3 course

10.1.Suggest six reasons why software dependability is important in most sociotechnical systems.



Six reasons why dependability is important are:

- Users may not use the system if they don't trust it.
- System failure may lead to a loss of business.
- An undependable system may lose or damage valuable data.
- An undependable system may damage its external environment.
- The reputation of the company who produced the system may be damaged hence affecting other systems.
- The system may be in breach of laws on consumer protection and the fitness of goods for purpose.

11.9 Explain how programming language support of exception handling can contribute to the reliability of software systems

What is exception? **Exception** is an event occurs during program execution and that event disturbs the normal execution flow of the program.

Exception handling: some programming languages support exception handling. When an exception occurs then through an object which contains information about exception and error, this object is sent to a method where we can run some other instructions according to requirements. So exception handling me technique are really helpful in many canario. Like we want to open a file that do not exist, we want to make a connection that can't be accessed, arithmetic error etc. Because of the exception handling mechanism we can effectively handles these kind of unexpected events like this it ensure the reliability of the software.



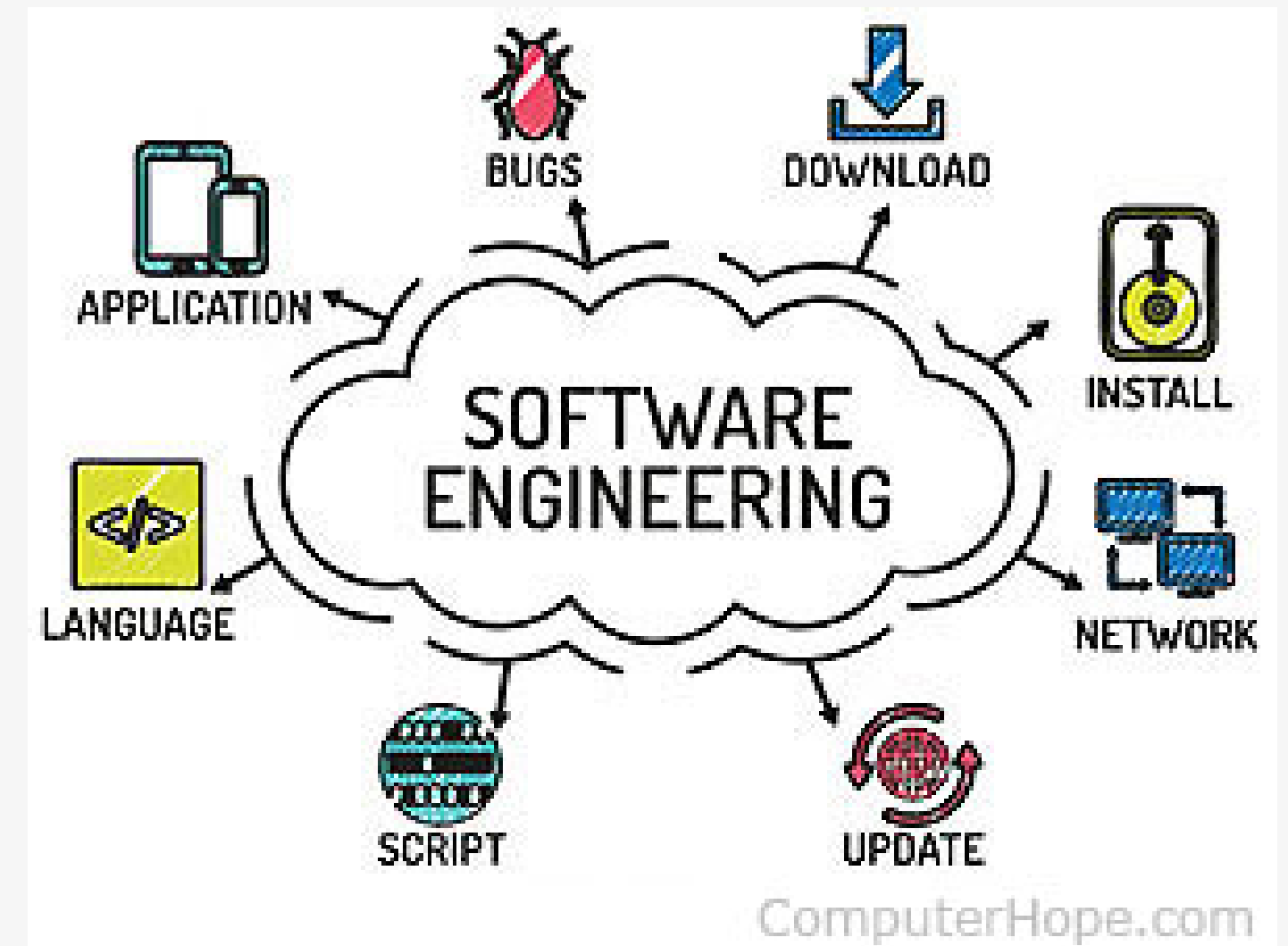
12.10 Should software engineers working on the specification and development of safety-related systems be professionally certified or licensed in some way? Explain your reasoning.

Typical formal definitions of **software engineering** are:

"research, design, develop, and test operating systems-level software, compilers, and network distribution software for medical, industrial, military, communications, aerospace, business, scientific, and general computing applications.

"the systematic application of scientific and technological knowledge, methods, and experience to the design, implementation, testing, and documentation of software" "the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software" "an engineering discipline that is concerned with all aspects of software production" and "the establishment and use of sound engineering principles in order to economically obtain software that is reliable and works efficiently on real machines."

The short version of the code summarizes aspirations at a high level of the abstraction; the clauses that are included in the full version give examples and details of how these aspirations change the way we act as software engineering professionals. Without the aspirations, the details can become legalistic and tedious; without the details, the aspirations can become high sounding but empty; together, the aspirations and the details form a cohesive code.



Software engineers shall commit themselves to making the analysis, specification, design, development, testing and maintenance of software a beneficial and respected profession. In accordance with their commitment to the health, safety and welfare of the public, software engineers shall adhere to the following Eight 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.

Computers have a central and growing role in commerce, industry, government, medicine, education, entertainment and society at large. Software engineers are those who contribute by direct participation or by teaching, to the analysis, specification, design, development, certification, maintenance and testing of software systems. Because of their roles in developing software systems, software engineers have significant opportunities to do good or cause harm, to enable others to do good or cause harm, or to influence others to do good or cause harm. To ensure, as much as possible, that their efforts will be used for good, software engineers must commit themselves to making software engineering a beneficial and respected profession. In accordance with that commitment, software engineers shall adhere to the following Code of Ethics and Professional Practice.

The Code contains eight Principles related to the behavior 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. The Principles identify the ethically responsible relationships in which individuals, groups, and organizations participate and the primary obligations within these relationships. The Clauses of each Principle are illustrations of some of the obligations included in these relationships. These obligations are founded in the software engineer's humanity, in special care owed to people affected by the work of software engineers, and the unique elements of the practice of software engineering

Ethical tensions can best be addressed by thoughtful consideration of fundamental principles, rather than blind reliance on detailed regulations. These Principles should influence software engineers to consider broadly who is affected by their work; to examine if they and their colleagues are treating other human beings with due respect; to consider how the public, if reasonably well informed, would view their decisions; to analyze how the least empowered will be affected by their decisions; and to consider whether their acts would be judged worthy of the ideal professional working as a software engineer. In all these judgments concern for the health, safety and welfare of the public is primary; that is, the "Public Interest" is central to this Code.