What is software engineering?

Software: code, documentation, user manuals, designs, specifications, test cases, plans and schedules

Engineering:

creatively applying science and math

to designing and building things

involves trade-offs, risk analysis, skill, and knowledge

A broad field that touches upon all aspects of developing and supporting a software system.

A discipline that deals with the building of software systems which are so large that they are built by a team or teams of engineers.

Multi-person construction of multi-version software.

A discipline whose aim is the production of fault-free software, delivered on-time and within budget, that satisfies the user’s needs. Furthermore, the software must be easy to modify when the user’s needs change.

Informatic 43 definition:

The process of constructing software

Phases of development other than programming

Principles and qualities of enduring value

The process of constructing software

Requirements

Design

Implementation

Verification

Maintenance

Principles and qualities of enduring value

Correctness

Adatability

Usability

Reliability

Modularity

Efficiency

Robustness

Software is everywhere.

No Sliver Bullet

Complexity

Each software is unique and complex in its own way.

Changeability

All successful software keeps changing.

Conformity

Conform to human needs and human needs are complex.

Invisibility

Software itself is invisibility.

“False” silver bullets

High-level languages

Time-sharing

Unified programming environments

OOP

Artificial intelligence

Potential silver bullets

Buy vs build

Requirements refinement and rapid prototyping

Great designers

Software engineering can be looked at different perspectives

Essential ingredients of software engineering

People, processes, tools

People: most important, most invested in

Principles of software engineering

make quality number 1

give products to customers early

understand the problem first

choose an appropriate process

good management is more important than good technology

people are the key to success

follow an architecture first process

use component based development to reduce coding effort

show the customer preliminary versions of the software frequently

have incremental releases

Rigor and formality

Separation of concerns

Modularity

Divide and conquer

Abstraction

Anticipation of change

Generality

Incrementality