SOFTWARE ENGINEERING

UNIT 1- MODULE 1 (INTRODUCTION)

Dr. Noopur Goel
Assistant Professor
Department of Computer Applications,
VBS Purvanchal University, Jaunpur, 222002
Email: noopurt11@gmail.com

TOPICS COVERED-

- Introduction to Software Engineering
- IEEE Definition of Software Engineering
- Need for Software Engineering/Why is Software Engineering Important?
- Definition of Software
- Program Vs Software Product
- Objectives of Software Engineering
- Essential Attributes of a Good Software
- Software Costs
- Software Project Failures
- Software Crisis
- Factors contributing to Software Crisis
- Software: An Art, Craft or Engineering?

Introduction to Software Engineering

- An engineering approach to develop software
  - With the support of
    - Defined set of rules
    - Methods
    - And tools
- Needs past experiences of developing software over the last several years
- Results in efficient and reliable software product
- Different from other engineering disciplines

IEEE Definition of Software Engineering

Software Engineering:

- is the application of a
  - systematic,
  - disciplined,
  - quantifiable approach
- to the development,
  - operation
  - and maintenance of software;
- that is the application of engineering to software.

Need For Software Engineering/ Why Is Software Engineering Important?

1. Individuals and society rely on advanced software systems. So need to be able to produce
   - reliable
   - and trustworthy systems
     - economically and quickly.
2. 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 systems,
   - **the majority of costs are the costs of changing the software after it has gone into use.**

Definition of Software

- Set of relevant software programs and associated documentation
- Software products may be developed for
  - A particular customer or
  - a general market

Program Vs Software Product:

1. A **program** is a set of instructions which is given to a computer in order to achieve a specific task whereas

   a **software** is a program which is made available for commercial business and is properly documented along with its licensing.

   Software = Program + documentation + licensing

2. A **program** is one of the stages involved in the development of the software, whereas

   A **software development** usually follows a life cycle, which involves

   - the feasibility study of the project,
   - requirement gathering,
   - development of a prototype,
   - system design,
   - coding and
   - testing.

Dual Role of Software:

1. **As a product –**
- It delivers the computing potential across network of Hardware.
- It enables the Hardware to deliver the expected functionality.
- It acts as information transformer because it produces, manages, acquires, modifies, displays, or transmits information.

2. As a vehicle for delivering a product –

- It provides system functionality (e.g., payroll system)
- It controls other software (e.g., an operating system)
- It helps build other software (e.g., software tools)

Objectives Of Software Engineering:

1. **Maintainability –** It should be feasible for the software to evolve to meet changing requirements.
2. **Correctness –** A software product is correct, if the different requirements as specified in the SRS document have been correctly implemented.
3. **Reusability –** A software product has good reusability, if the different modules of the product can easily be reused to develop new products.
4. **Testability –** software facilitates both the establishment of test criteria and the evaluation of the software with respect to those criteria.
5. **Reliability –** It is an attribute of software quality. The extent to which a program can be expected to perform its desired function, over an arbitrary time period.
6. **Portability –** In this case, software can be transferred from one computer system or environment to another.
7. **Adaptability –** In this case, software allows differing system constraints and user needs to be satisfied by making changes to the software.

Essential Attributes of a Good Software

1. **Maintainability** : a critical attribute because **software change** is unavoidable due to changing business environment. Hence, software should be written in such a way so that it can evolve to meet the changing needs of customers.
2. **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.
3. **Efficiency:** Software should not make wasteful use of system resources such as memory and processor cycles. Efficiency therefore includes **responsiveness, processing time, memory utilization, etc**.
4. **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.

Software Costs

- **Software costs** are usually higher than the **hardware costs**.
- In case of software for a log run, **maintenance cost** of a software **is higher than** the **cost of developing it**.
- However, software engineering is concerned with the **cost-effective software development.**

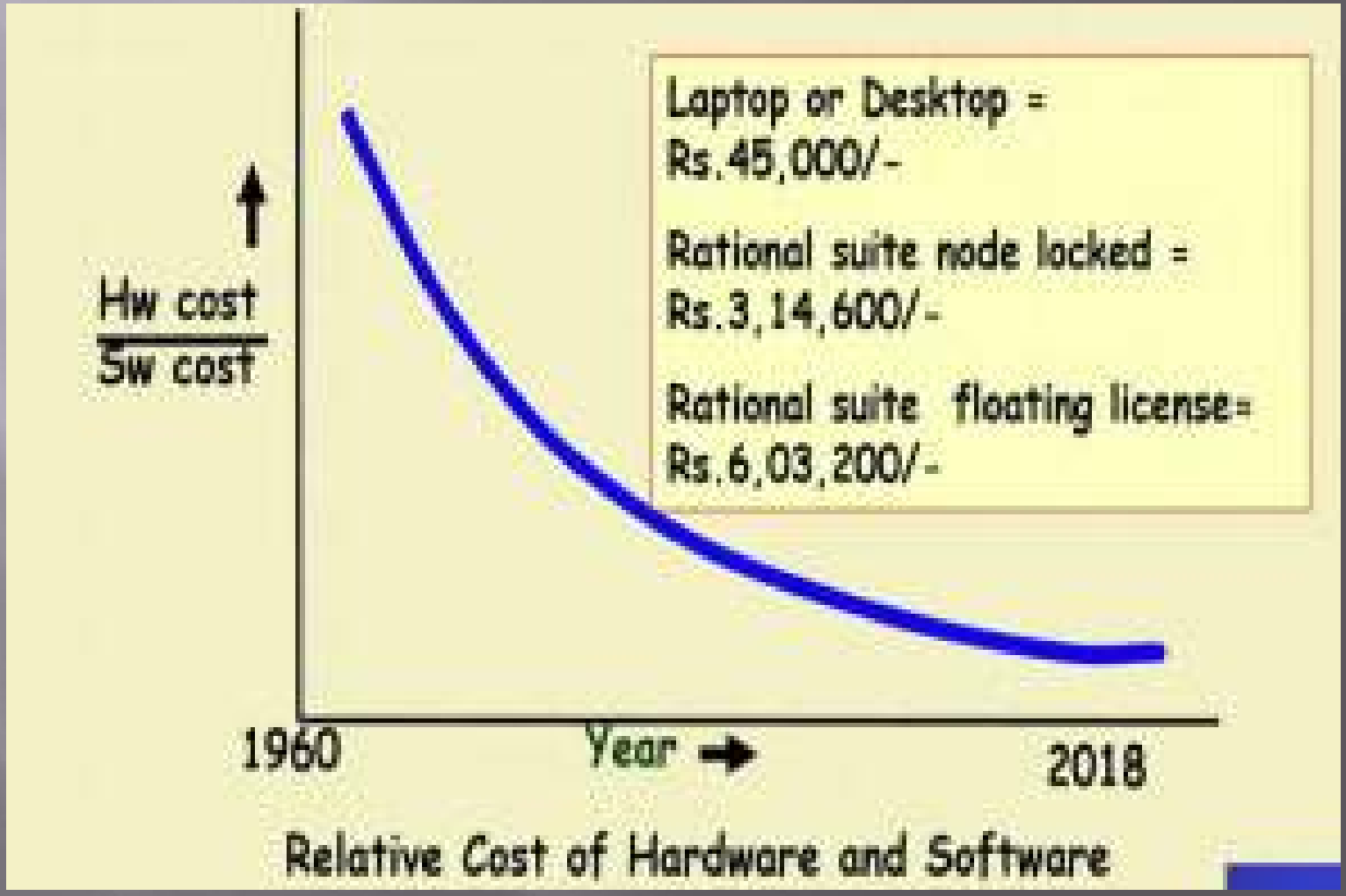
Software Project Failures

- **Due to Increasing System Complexity**
  - ➢As Changing demands with time;
  - ➢Software to be developed and delivered faster;
  - ➢More complex and large systems are required with new capabilities;
- **Due to failure to use software engineering methods**
  - ➢computer programs are very easy written without using software engineering methods and techniques.
  - ➢Many software companies do not follow software engineering methods to develop software products which results in more expensive and less reliable software.

Software Crisis

- There are times when software products
  - Fail to meet user requirements.
  - Expensive.
  - Difficult to modify, debug, and augment
  - Often does not meet time-to-delivery.
  - Use resources non-optimally.

Software Crisis (Continued)

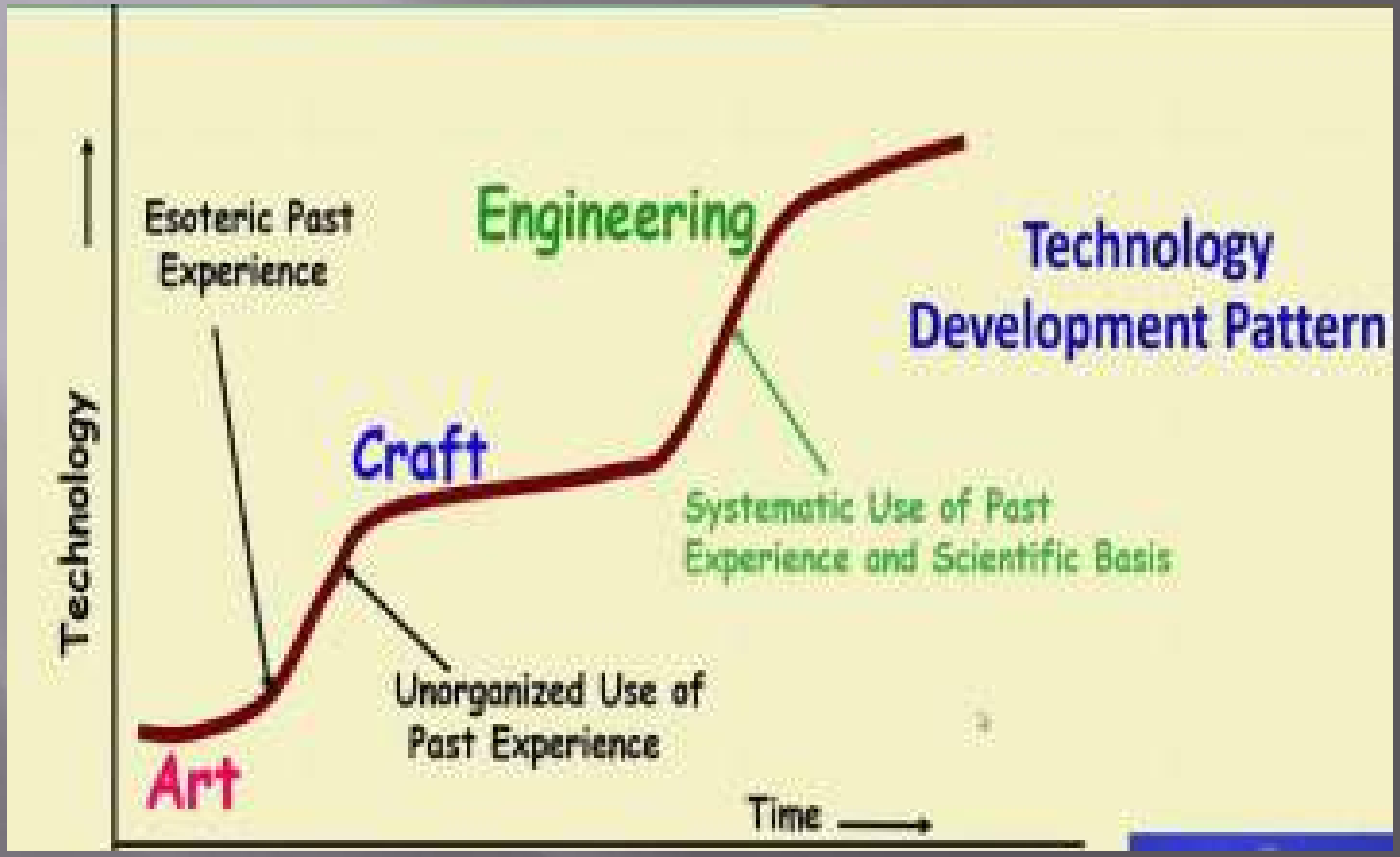


Software Crisis (Continued)

Factors contributing to Software Crisis

- Larger and more complex problems,
- Poor project management because based on manual efforts,
- Lack of adequate training of developers in latest software engineering techniques.
- Increasing skill shortage over the years
- Low productivity developments compared to the increasing problem sizes.

Software: An Art, Craft or Engineering?



REFERENCES-

1. SOFTWARE ENGINEERING, Ian Sommerville, Ninth Edition,Addison-Wesley, Pearson
2. Fundamentals of Software Engineering, Rajib Mall, PHI Learning
3. .https://www.geeksforgeeks.org/software-engineering-introduction-to-software-engineering/
4. https://www.tutorialandexample.com/software-engineering-tutorial/