Software Engineering – ESC501

- Prof. Poulami Dutta

DEPARTMENTAL VISION

- To be a leader in Computer Science and Engineering education by providing a platform to produce industry and/or research oriented individuals contributing to the enrichment of the society.

DEPARTMENTAL MISSION

- To impart quality education by applying ingenious and modern methods of pedagogy thereby calibrating one's individual outlook towards problem solving.
- To recognize the flair and talent of individuals who will be nurtured to become leaders and innovators in industry and education and bring them to the limelight by enhancing their entrepreneurship skills.
- To promote higher studies and research activities by indulging in innovative projects and collaborative ventures with the industry and premier institutes.
- To create graduates to be successful, ethical and lifelong learners by imbibing holistic education to promote sustainability and contribute to the social well-being.
- To boost employability skills through intra, inter-departmental and inter-institutional activities beyond curriculum thereby invigorating team-building activities and leadership skills to instill confidence and creativity.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

- **PEO1:** To excel as successful career professionals in emerging fields of Computer Science and Engineering and to pursue research.
- **PEO2:** To establish expertise in solving contemporary problems in analysis, design and evaluation using modern tools and technologies.
- **PEO3:** To engage in lifelong learning and professional development to adapt to rapidly changing work environments.
- **PEO4:** To demonstrate entrepreneurial and managerial skills to make fruitful contributions towards overall sustainable societal development.

PROGRAM OUTCOMES (POs)

- **PO1. Engineering Knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and engineering specialization to the solution of complex engineering problems.
- **PO2. Problem analysis:** Identify, formulate, research literature, and analyze engineering problems to arrive at substantiated conclusions using first principles of mathematics, natural and engineering sciences.
- **PO3. Design/Development of solutions:** Design solutions for complex engineering problems and design system components, processes to meet the specifications with consideration for the public health and safety and the cultural societal and environmental considerations.
- **PO4. Conduct investigations of complex problems:** Use research based knowledge including design of experiments, analysis and interpretation of data and synthesis of the information to provide
- **PO5. Modern tool usage:** Create, select and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
- **PO6. The engineer and society:** Apply reasoning informed by the contextual knowledge to access societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
- **PO7. Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of and need for sustainable development.
- **PO8. Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
- **PO9. Individual and team work:** Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.
- **PO10. Communications:** Communicate effectively with the engineering community and with the society at large. Be able to comprehend and write effective reports documentation. Make effective presentations and give and
- **PO11. Project management and finance:** Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team. Manage projects in multidisciplinary environments.
- **PO12. Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

- **PSO1:** Ability to develop the solutions for scientific, analytical and other complex problems in the area of Computer Science and Engineering.
- **PSO2:** Ability to apply suitable problem solving skill integrated with professional competence to develop solutions catering to the industry, research and societal needs in the field of Computer Science and Engineering and its allied areas.

Course Outcomes

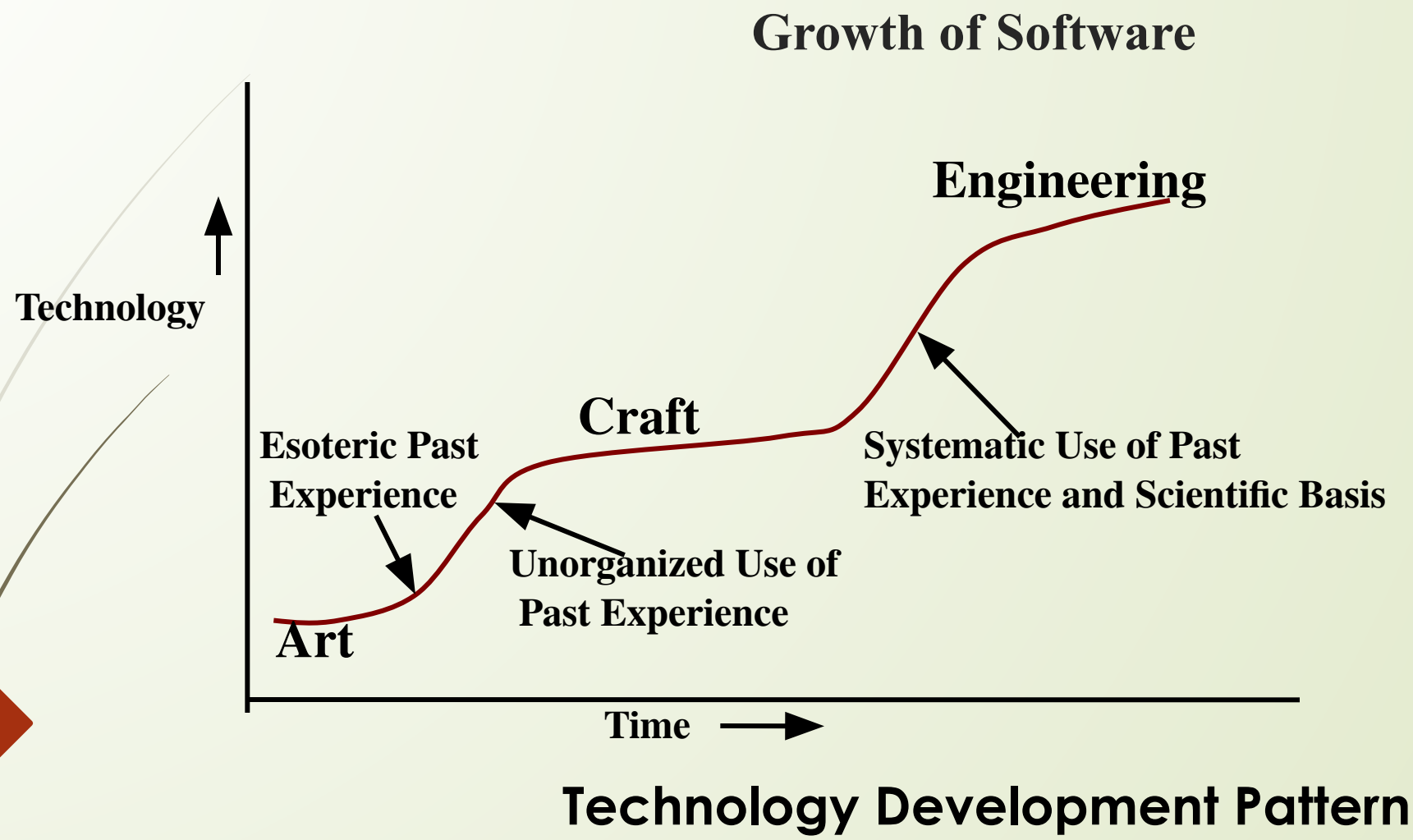
ESC-501.1	Identify a suitable life cycle model that meets specification, performance, maintenance and quality requirements for a given software development problem.
ESC-501.2	Construct a design model using appropriate software engineering methodologies from the given specifications.
ESC-501.3	Analyze software requirements through a productive working relationship with various stakeholders of the project to come up with a viable project plan to deliver the software product with optimized size, effort, time and cost using suitable estimation methods.
ESC-501.4	Develop a working model to monitor the progress of the project and ensure that it provides quality assurance in the short and long run by utilizing relevant standards and suitable testing practices.
ESC-501.5	Design software products/prototypes related to various real life/IT problems by relating the software architectural styles.

ILOs (Intended Learning Outcomes)

- Identify the scope and necessity of software engineering.
- Identify the causes of and solutions for software crisis.
- Differentiate a piece of program from a software product.

What is software engineering?

- Software engineering is composed of two words, **software** and **engineering**.
- **Software** is defined as a collection of programs, procedures, rules, data and associated documentation. The s/w is developed keeping in mind certain h/w and operating system consideration commonly known as platform.
- **Engineering** means systematic procedure to develop software.
- **Software engineering** as an engineering branch is associated with the development of software product using well-defined scientific principles, methods and procedures. The outcome of software engineering is an efficient and reliable software product.
- **IEEE Definition:** The application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software.



Technology Development Pattern

Why Study Software Engineering?

- **To acquire skills to develop large programs:**
  - Exponential growth in complexity and difficulty level with size.
  - The ad hoc approach breaks down when size of software increases.
- **Ability to solve complex programming problems:**
  - How to break large projects into smaller and manageable parts?
- **Learn techniques of:**
  - specification, design, interface development, testing, project management, etc.
- **To acquire skills to be a better programmer:**
  - Higher Productivity
  - Better Quality Programs

Software Quality

- **Portability:** to be able to work in different operating system environments, in different machines, with other software products.
- **Usability:** if different categories of users (i.e. both expert and novice users) can easily invoke the functions of the product, the software is said to be usable.
- **Reusability:** if different modules of the product can easily be reused to develop new products, the software is said to be reusable.
- **Correctness:** if different requirements as specified in the SRS document have been correctly implemented, the software is said to be correct.
- **Maintainability:** if errors can be easily corrected, new functions can be easily added to the product, and the functionalities of the product can be easily modified, etc. the software is said to be maintainable.

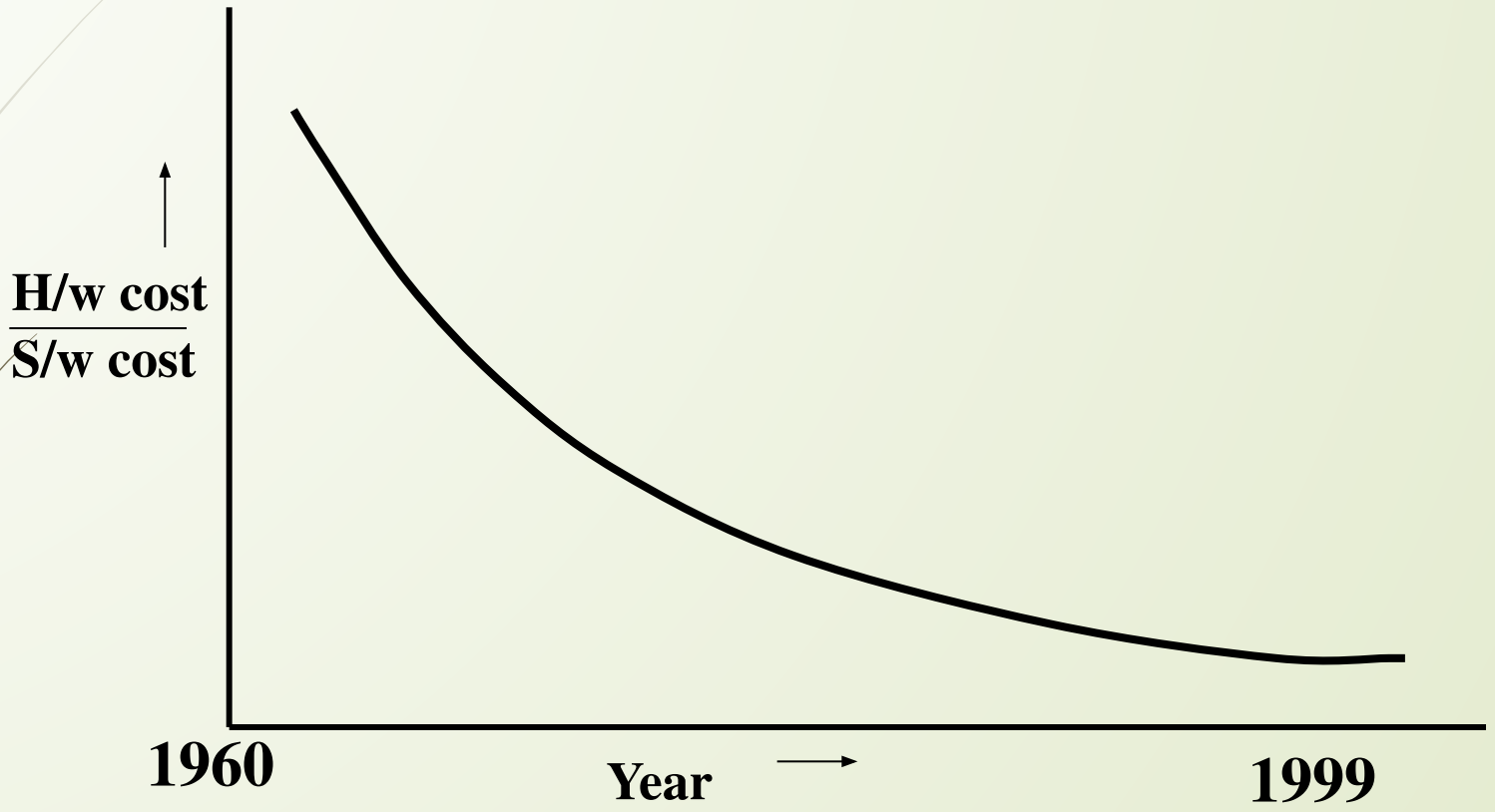
Types of Software

- **System Software:** It includes the operating system & all the utilities to enable the computer to run. Ex-OS.
- **Application Software:** It consists of programs to perform user oriented tasks. Ex-word processor, database management.

Software Crisis

- **Software products:**
  - fail to meet user requirements.
  - frequently crash.
  - expensive.
  - difficult to alter, debug, and enhance.
  - often delivered late.
  - use resources non-optimally.

Software Crisis (cont.)



Relative Cost of Hardware and Software

Factors contributing to the software crisis

- Larger problems,
- Lack of adequate training in software engineering,
- Increasing skill shortage,
- Low productivity improvements.

Need for Software Engineering

- **Large Software** – It is easier to build a wall than a house or a building.
- **Scalability** – Program with 100 LOC and 10000 LOC.
- **Cost** – Hardware cost has come down but software cost continues to be on the rise.
- **Dynamic Nature** – New enhancements being done to an already existing software.
- **Quality Management** - Better process of software development provides better and quality software product.

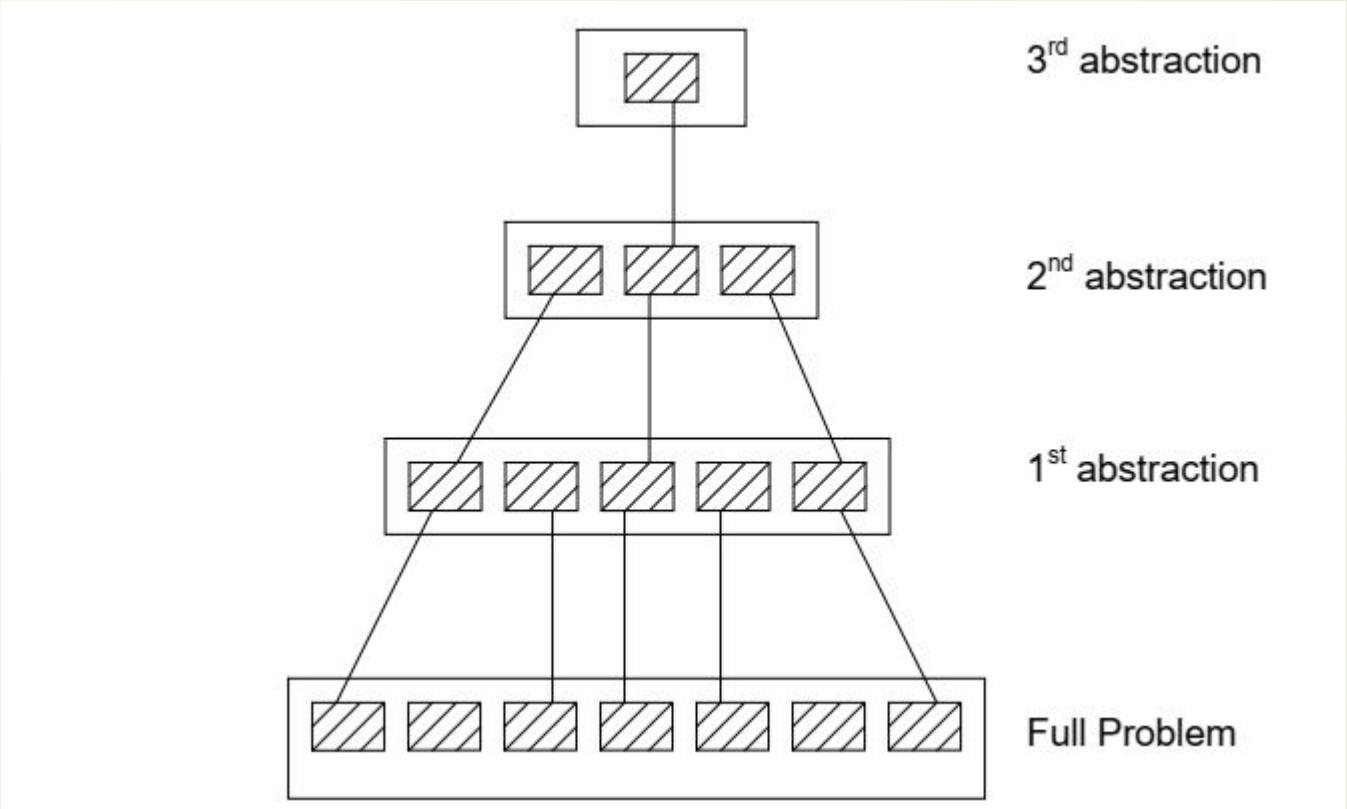
Programs versus Software Products

- Set of instructions related to each other
- Usually small in size
- Author himself is sole user
- Single developer
- Programs are defined by individuals for their personal use
- Lacks proper user interface
- Lacks proper documentation
- Ad hoc development
- Limited functionality

- Collection of programs designed for specific task
- Large in size
- Large number of users
- Team of developers
- A software product is usually developed by a group of engineers working as a team
- Well-designed interface
- Well documented & user-manual prepared
- Systematic development using principles of SE
- Exhibits more functionality

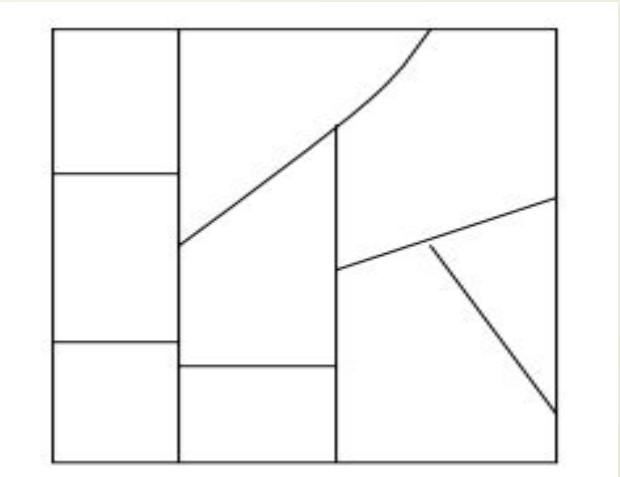
Software Engineering Principles

- **Abstraction** – Omits irrelevant details and consider aspects that are relevant.



Software Engineering Principles (cont.)

- **Decomposition** – A complex problem is divided into several smaller problems and then the smaller problems are solved one by one.
  - Each component is solved independently.



Emergence of Software Engineering

- Assembly Language
- High-Level Language
- Control Flow-based Design
- Data-Structure Oriented Design
- Data-Flow Oriented Design
- Object-Oriented-based Design

- Structured vs. Unstructured Programming


Thank you!