

BIT475CO

SOFTWARE ENGINEERING



UNIT-1

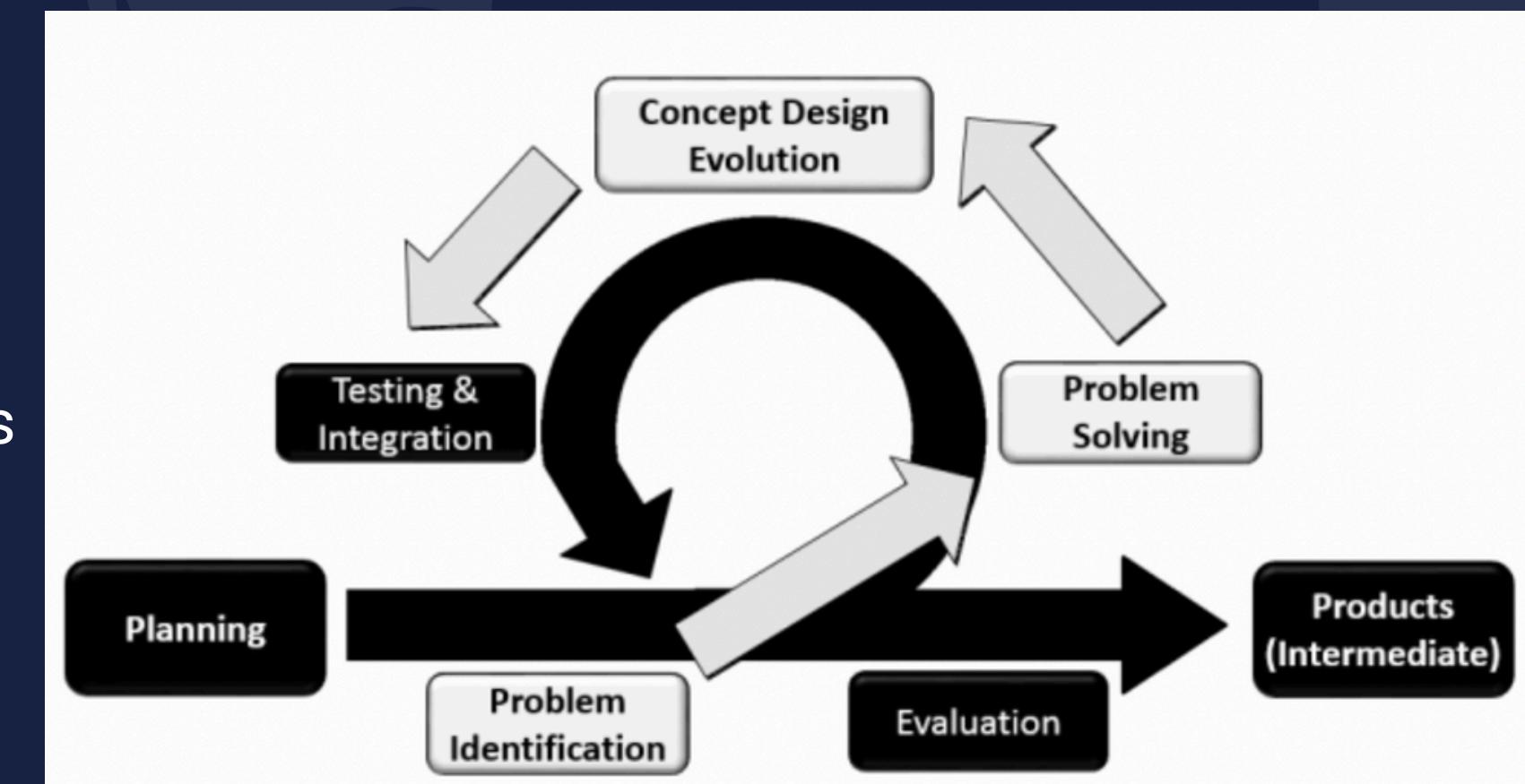
INTRODUCTION TO SOFTWARE ENGINEERING

TOPICS TO BE DISCUSSED

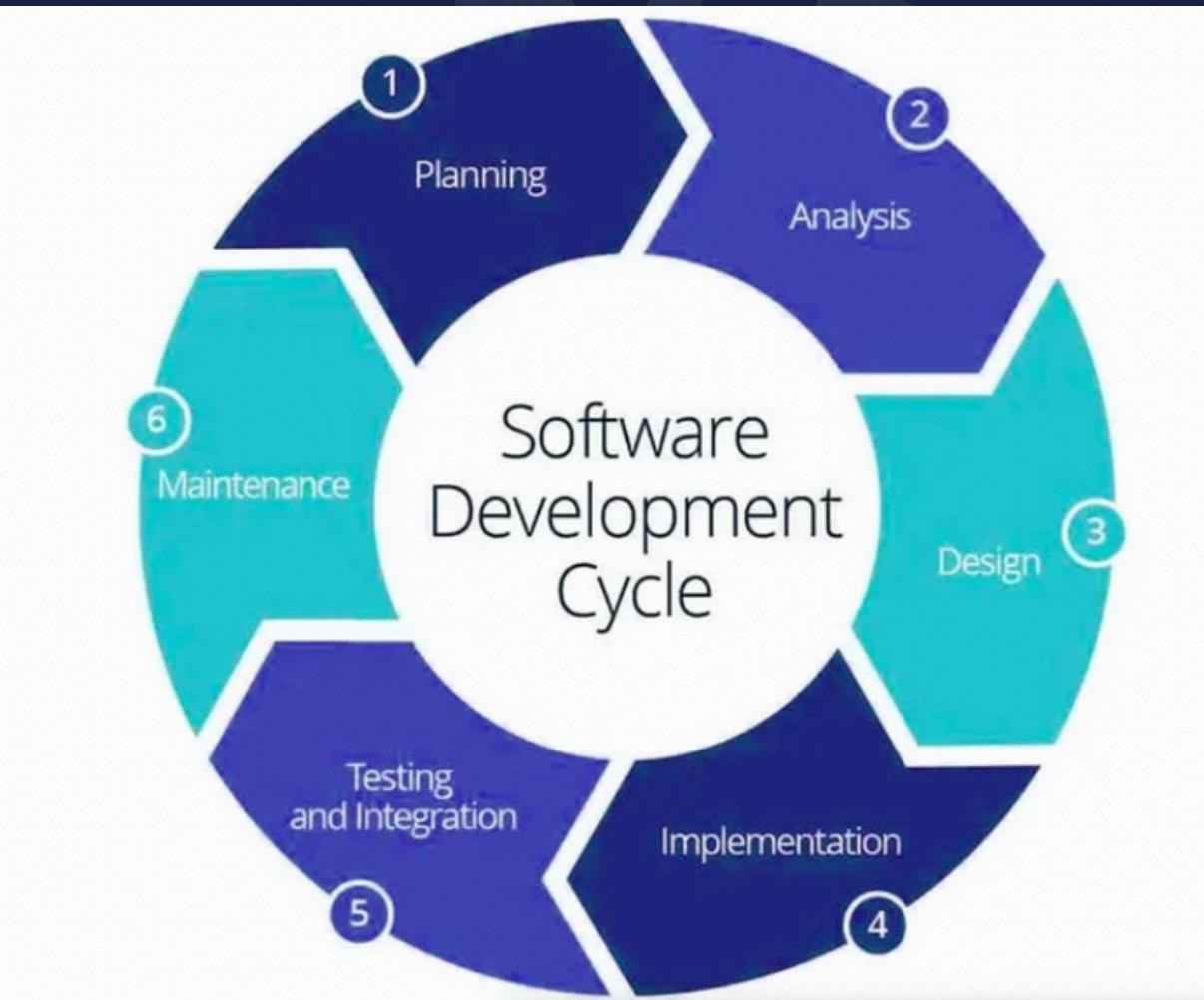
- Definition of software engineering
- the evolving role of software
- changing nature of software
- characteristics of software
- a generic view of software engineering
- software engineering-layered technology

SOFTWARE ENGINEERING

- Software engineering is the systematic application of engineering approaches to the development, operation, and maintenance of software.
- Software engineering principles guide the systematic design, development, and maintenance of the application to ensure reliability, security, and scalability.
- Engineers follow a structured process, including requirements gathering, design, implementation, testing, and maintenance, to deliver a high-quality and user-friendly application.
- It involves the application of principles from engineering, computer science, and project management to develop high-quality software products efficiently.
- Software engineering is concerned with theories, methods and tools for professional software development.



- Software engineering is concerned with theories, methods and tools for professional software development.
- Software engineering involves wider responsibilities than simply the application of technical skills.
- Software engineering is an engineering discipline that is concerned with all aspects of software production from the early stages of system specification through to maintaining the system after it has gone into use.
- Using appropriate theories and methods to solve problems bearing in mind organizational and financial constraints.
- Not just technical process of development. Also project management and the development of tools, methods etc. to support software production.



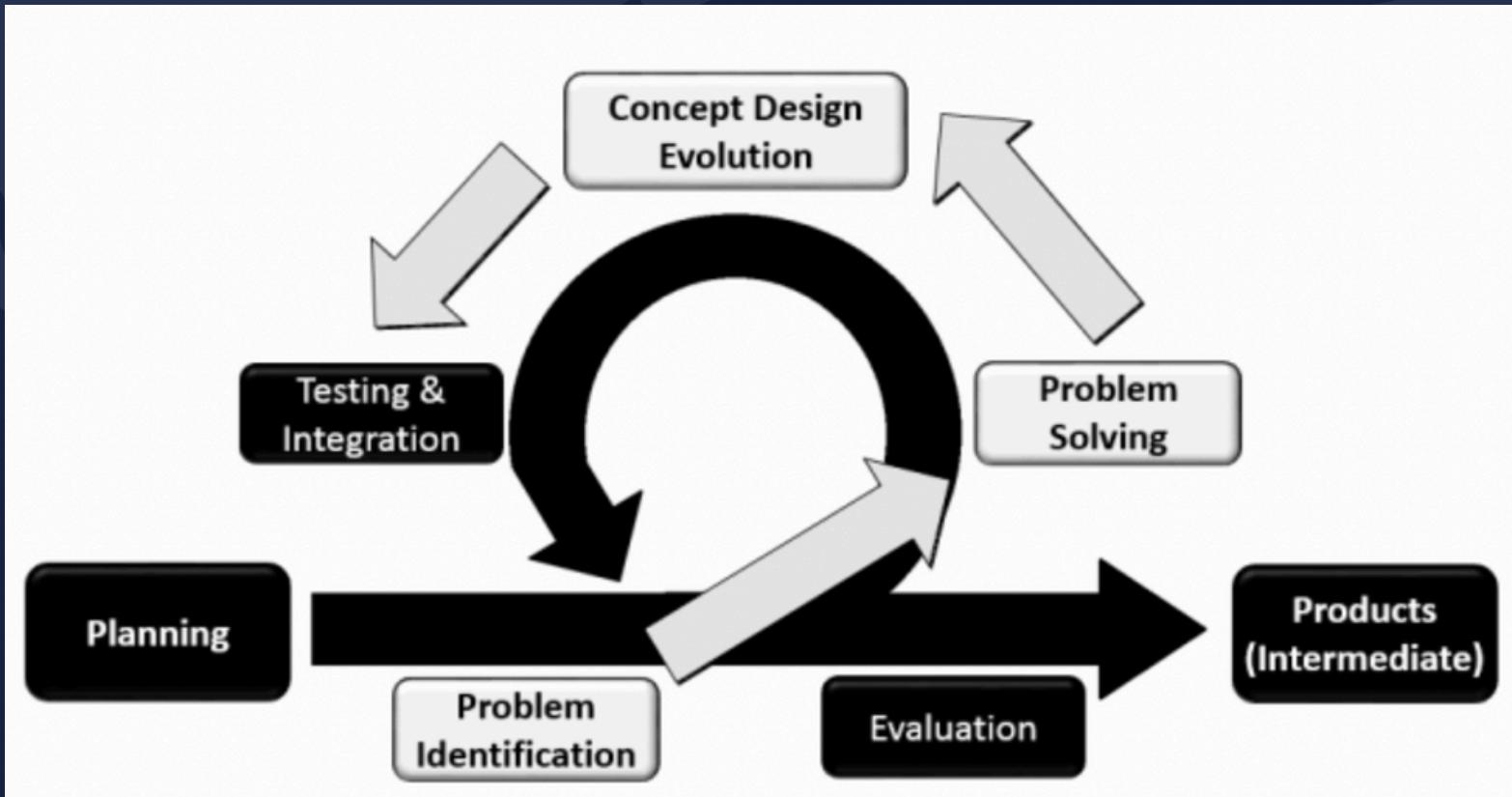
THE EVOLVING ROLE OF SOFTWARE

- Software has become pervasive in modern society, impacting various aspects of our lives.
- The use of software has transformed each and every industry, providing convenient and efficient ways for people to deal with daily chores.
- From embedded systems in everyday devices to complex applications powering businesses, the role of software continues to evolve rapidly.



THE EVOLVING ROLE OF SOFTWARE

- Software has become pervasive in modern society, impacting various aspects of our lives.
- The use of software has transformed each and every industry, providing convenient and efficient ways for people to deal with daily chores.
- From embedded systems in everyday devices to complex applications powering businesses, the role of software continues to evolve rapidly.



CHANGING NATURE OF SOFTWARE

- Software is not static; it evolves and adapts over time.
 - Example: Social media platforms
- Evolution driven by changing requirements, technological advancements, and user feedback.
 - Example: Video streaming services
- Emphasis on iterative development and continuous improvement.
 - Example: Software development teams practicing Agile methodologies release incremental updates to their products frequently, allowing for rapid adaptation to changing requirements and market conditions.

IMPORTANCE OF SOFTWARE ENGINEERING

Reduces Complexity:

- Big software is always complicated and challenging to progress.
- Software engineering has a great solution to reduce the complication of any project.
- Software engineering divides big problems into various small issues, and then start solving each small issue one by one. All these small problems are solved independently to each other.

Minimize Software Cost:

- The software needs a lot of hard work and software engineers are highly paid experts.
- A lot of manpower is required to develop software with a large number of codes.
- But in software engineering, programmers project everything and decrease all those things that are not needed.
- In turn, the cost for software productions becomes less as compared to any software that does not use software engineering method.

IMPORTANCE OF SOFTWARE ENGINEERING

Decrease Time:

- Anything that is not made according to the project always wastes time. And if we are making great software, then we may need to run many codes to get the definitive running code.
- This is a very time-consuming procedure, and if it is not well handled, then this can take a lot of time.
- So if we are making our software according to the software engineering method, then it will decrease a lot of time.

Reliable Software:

- The software should be secure, means if we have delivered the software, then it should work for at least its given time or subscription.
- And if any bugs come in the software, the company is responsible for solving all these bugs.
- Because in software engineering, testing and maintenance are given, so there is no worry of its reliability.

IMPORTANCE OF SOFTWARE ENGINEERING

Handling Big Projects:

- Big projects are not done in a couple of days, and they need lots of patience, planning, and management.
- And to invest six and seven months of any company, it requires heaps of planning, direction, testing, and maintenance.
- No one can say that they have given four months of a company to the task, and the project is still in its first stage.
- Because the company has provided many resources to the plan and it should be completed.
- So to handle a big project without any problem, the company has to go for a software engineering method.

IMPORTANCE OF SOFTWARE ENGINEERING

Effectiveness:

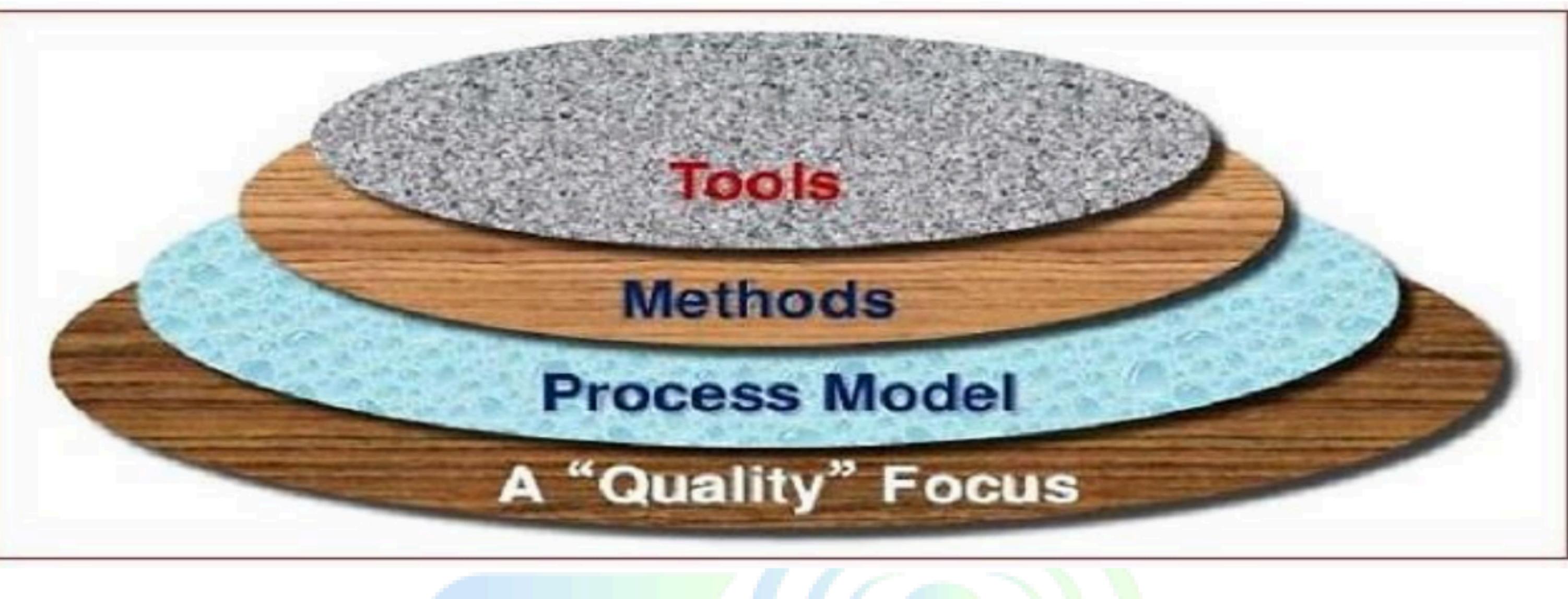
- Effectiveness comes if anything has made according to the standards.
- Software standards are the big target of companies to make it more effective.
- So Software becomes more effective in the act with the help of software engineering.

ROLE OF SOFTWARE ENGINEERING

- Confer with systems analysts, engineers, programmers and others to design system and to obtain information on project limitations and capabilities, performance requirements and interfaces.
- Modify existing software to correct errors, allow it to adapt to new hardware, or to improve its performance.
- Analyze user needs and software requirements to determine the feasibility of design within time and cost constraints.
- Consult with customers about software system design and maintenance.
- Coordinate software system installation and monitor equipment functioning to ensure specifications are met.
- Design, develop and modify software systems, using scientific analysis and mathematical models to predict and measure outcome and consequences of design.
- Develop and direct software system testing and validation procedures, programming, and documentation.
- Analyze information to determine, recommend, and plan computer specifications and layouts, and peripheral equipment modifications.

Software Development is a Layered Technology:

Software development is totally a **layered technology**. That means, **to develop software** one will have to go from **one layer to another**. The layers are related and each layer demands the fulfilment of the previous layer. The figure below is the **upward flowchart of the layers** of software development.



1. A Quality Focus:

Software engineering must rest on an **organizational commitment** to quality. **Total quality management** and similar **philosophies** foster a continuous process **improvement culture** and this **culture ultimately leads** to the development of increasingly more mature approaches to **software engineering**. The bedrock that supports software engineering is a **quality focus**.

2. Process:

The **foundation** for **software engineering** is the **process layer**. The **process** defines a **framework** for a set of **Key Process Areas (KPAs)** that must be established for the effective delivery of **software engineering technology**.

This establishes the context in which technical methods are applied, work products such as **models, documents, data, reports, forms, etc.** are produced, milestones are established, quality is ensured, and change is properly managed.

3. Methods:

Software engineering methods provide the **technical how-to** for building software. **Methods** will include **requirements analysis, design, program construction, testing, and support**. This relies on a set of basic principles that govern each area of the technology and include **modelling activities** and other **descriptive techniques**.

4. Tools:

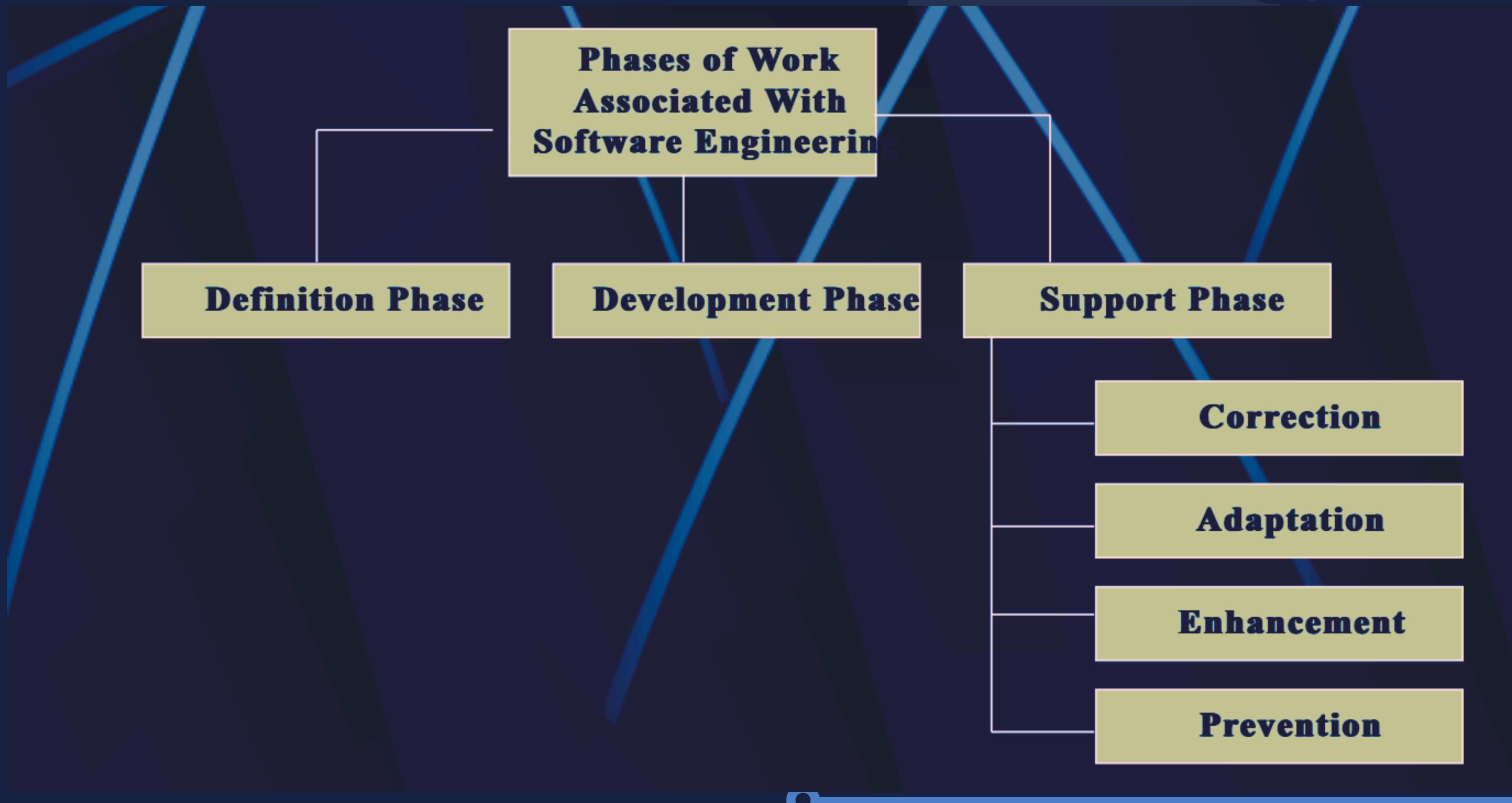
Software engineering tools provide **automated** or **semi-automated** support for the **process and the methods**. When **tools** are **integrated** so that information created by **one tool can be used by another**, a system for the support of software development called **computer-aided software engineering** is established.

GENERIC VIEW OF SOFTWARE ENGINEERING

The work associated with Software Engineering can be categorized in to three phases regardless of application area, project size or complexity;

- Definition phase
- Development phase
- Support phase

GENERIC VIEW OF SOFTWARE ENGINEERING



GENERIC VIEW OF SOFTWARE ENGINEERING

Definition phase:

- focus on what where software engineer focus on What information to process.
- What function and performance are desired.
- What system behavior can be expected.
- What interfaces to be established.
- What design constraints exist.
- What validation criteria are required.

In definition phase the key requirements of system and software are identified, three major tasks will occur:

- System or information engineering
- Software project planning
- Requirements analysis.

GENERIC VIEW OF SOFTWARE ENGINEERING

Development phase:

- focus on how data are to be structured
- How functions are to be implemented. how procedural details are to be implemented
- How interfaces are to be characterized
- How the design to be translated to programs
- How testing to be performed.

In Development phase three specific technical task should always occur:

- Software design
- Code generation
- Software testing.

GENERIC VIEW OF SOFTWARE ENGINEERING

Support Phase:

- focus on change associated with:
 - Error Correction.
 - Adaptation required as software environment evolved.
 - Enhancement.

In Support Phase four types of changes are encountered:

- Correction of undiscovered defect.
- Adaptation for new environment.
- Enhancement for new features required by customer.
- Prevention by making change to computer program to be more easily corrected, adapted and enhanced.

TYPES OF SOFTWARE

- There are many different types of software. Software can be categorized to: -
 - Custom Software
 - Generic Software
 - Embedded Software

TYPES OF SOFTWARE

- **Custom Software:** Software that is developed to meet specific need of a particular customer and tends to be little use to others. For example websites, air traffic control, software to manage specialize organization need
- **Generic Software:** Software that is designed to be sold in open market, perform function that many people need, and run on a general purpose computers. Example(word processor, spreadsheets, compiler web browsers, operating systems, etc
- **Embedded Software:** Software that is run on specific hardware devices. These devices are intended to be sold on open market. Examples: DVD players, Washing machine, Microwave oven, etc.

TYPES OF SOFTWARE

Other categorization of software

1. System Software.
2. Real-time Software.
3. Business Software.
4. Engineering and Scientific Software.
5. Embedded Software.
6. Web-based Software.
7. Artificial Intelligence Software