Introduction

Definition of Software

Types of Software

Characteristics of Software

Attributes of Good Software

Definition of software engineering

Software Engineering Costs

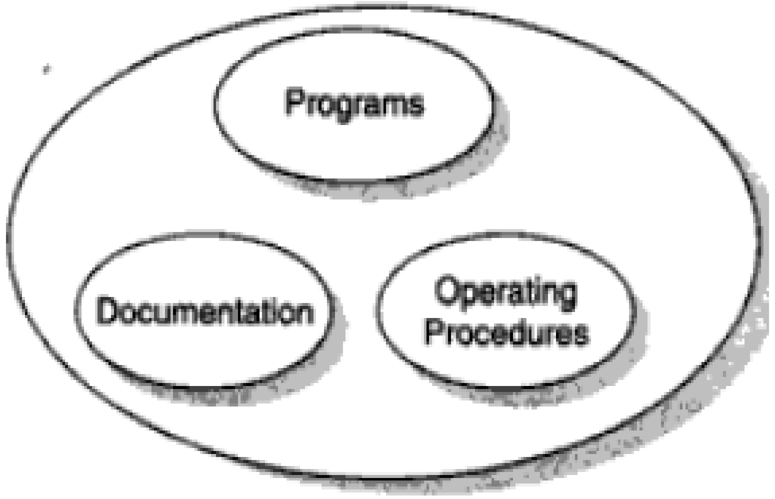
Key challenges that software engineering facing

System Engineering and Software Engineering

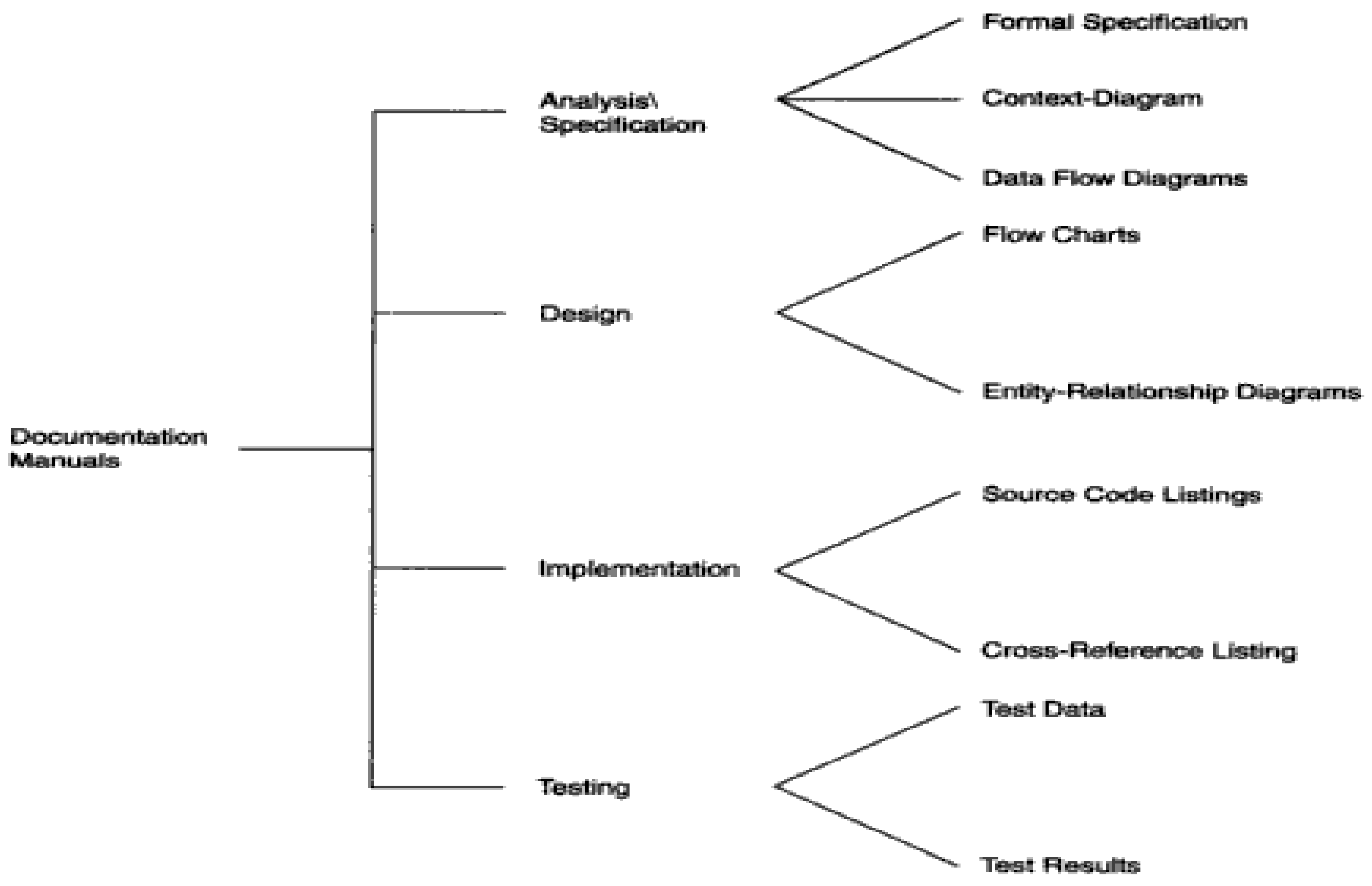
Professional Practices

Definition of Software

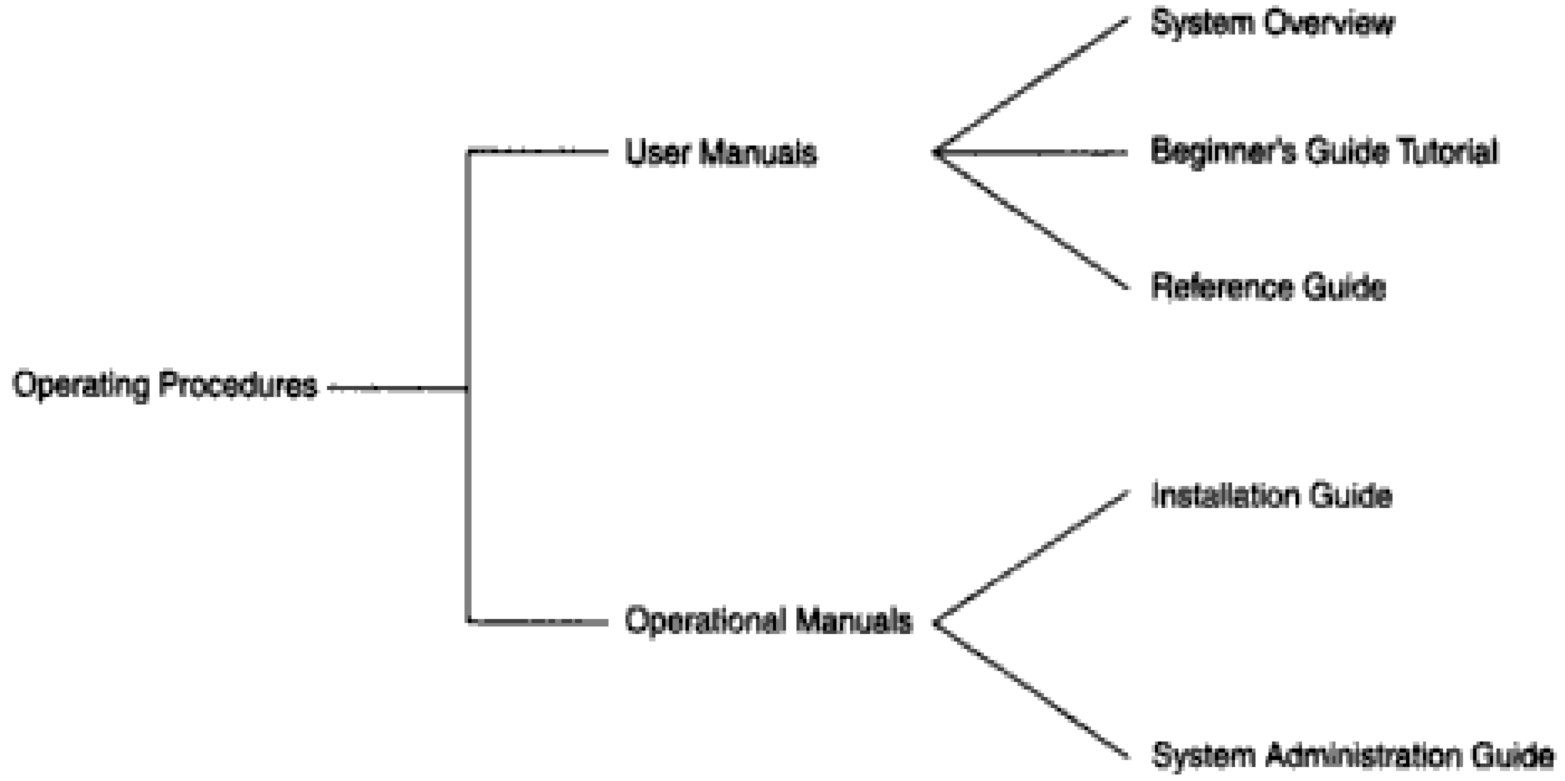
- Software is more than programs. It consists of program, documentation of any facet of the program and the procedures used of setup and operate the software system.
- In brief Software is:
  i. instructions (computer programs that when executed provide desired function and performance,
  ii. data structures that enable the programs to adequately manipulate information, and
  iii. documents that describe the operation and use of programs.
- Software= Program + Documentation + Operating Procedures
- Any program is a subset of software and it becomes software only if documentation and operating procedure manuals are prepared.
- Program is a combination of source code and object code. Documentation consists of different types of manuals



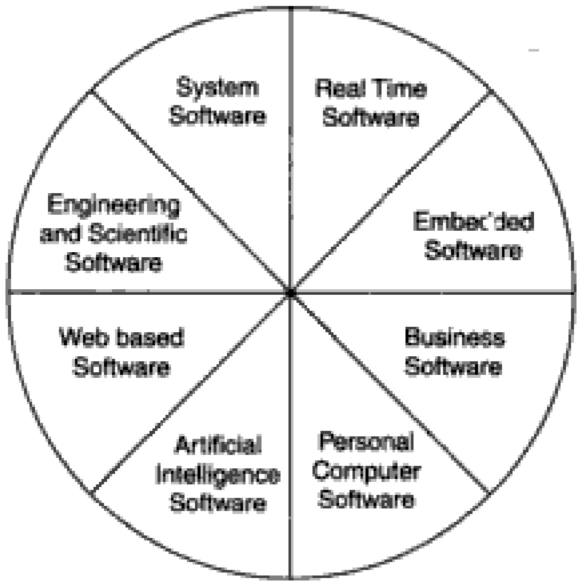
- Documentation consists of different types of manuals as shown in figure:



- Operating procedures consist of instructions to setup and use the software system and instructions on how to react to system failure.
- List of operating procedure manuals/documentation is given in figure.



Types of Software


System Software
Real Time Software
Engineering and Scientific Software
Embec'ded Software
Web based Software
Business Software
Artificial Intelligence Software
Personal Computer Software

Changing nature of Software/Types of Software

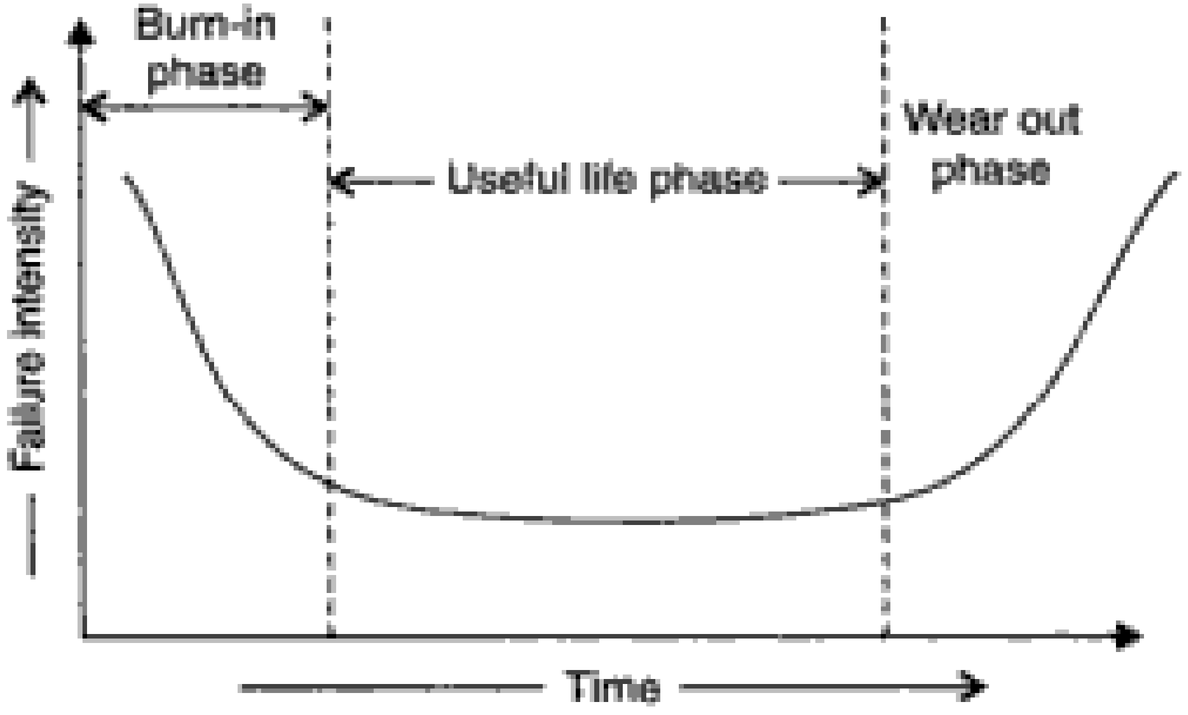
- The nature of software has changed a lot over the years.
- **System software:**
  - Infrastructure software come under this category like compilers, operating systems, editors, drivers, etc.
  - Basically system software is a collection of programs to provide service to other programs.
- **Real time software:**
  - These software are used to monitor, control and analyze real world events as they occur.
  - An example may be software required for weather forecasting.
  - Such software will gather and process the status of temperature, humidity and other environmental parameters to forcast the weather.
- **Embedded software:**
  - This type of software is placed in "Read-Only-Memory (ROM)"of the product and control the various functions of the product.
  - The product could be an aircraft, automobile, security system, signaling system, control unit of power plants, etc.
  - The embedded software handles hardware components and is also termed as intelligent software.
- **Business software:**
  - This is the largest application area.
  - The software designed to process business applications is called business software.
  - Business software could be payroll, file monitoring system, employee management, account management.
  - It may also be a data warehousing tool which helps us to take decisions based on available data.
  - Management information system, enterprise resource planning (ERP) and such other software are popular examples of business software.
- **Personal computer software:**
  - The software used in personal computers are covered in this category.
  - Examples are word processors, computer graphics, multimedia and animating tools, database management, computer games etc.
  - This is a very upcoming area and many big organizations are concentrating their effort here due to large customer base.
- **Artificial intelligence software:**
  - Artificial Intelligence software makes use of non numerical algorithms to solve complex problems that are not amenable to computation or straight forward analysis.
  - Examples are expert systems, artificial neural network,signal processing software etc.
- **Web based software:**
  - The software related to web applications come under this category.
  - Examples are CGI, HTML, Java, Perl, DHTML etc.
- **Other Software**
  - Entertainment software like games software,
  - Scientific software like matlab, autocad
  - Utility Software like antivirus software
- Now the trends has emerged to provide source code to the customer and organizations known as open source software
- For example : LINUX, MySQL, PHP, Open office, Apache webserver etc.

Software Products

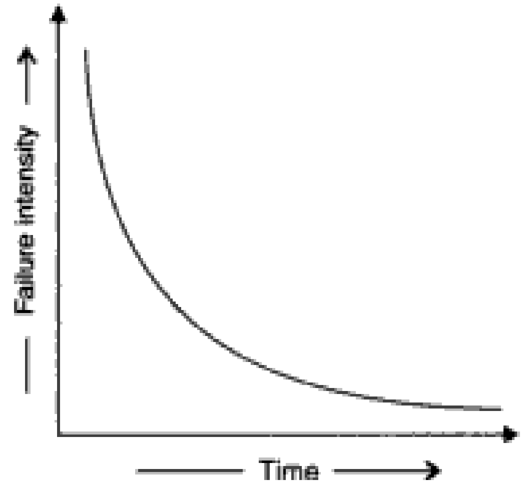
- **Software Products** are nothing but software systems delivered to the customer with the documentation that describes how to install and use the system.
- In certain cases, software products may be part of system products where hardware, as well as software, is delivered to a customer.
- Software products are produced with the help of the software process. The software process is a way in which we produce software.
- Can be of two types
  - Generic products
    - Stand-alone systems that are marketed and sold to any customer who wishes to buy them.
    - The specification of what the software should do is owned by the software developer and decisions on software change are made by the developer.
    - Examples – Operating system,PC software such as graphics programs, project management tools; CAD software; software for specific markets such as appointments systems for dentists.
  - Customized products
    - Software that is commissioned by a specific customer to meet their own needs.
    - The specification of what the software should do is owned by the customer for the software and they make decisions on software changes that are required.
    - Examples – embedded control systems, air traffic control software, traffic monitoring systems.

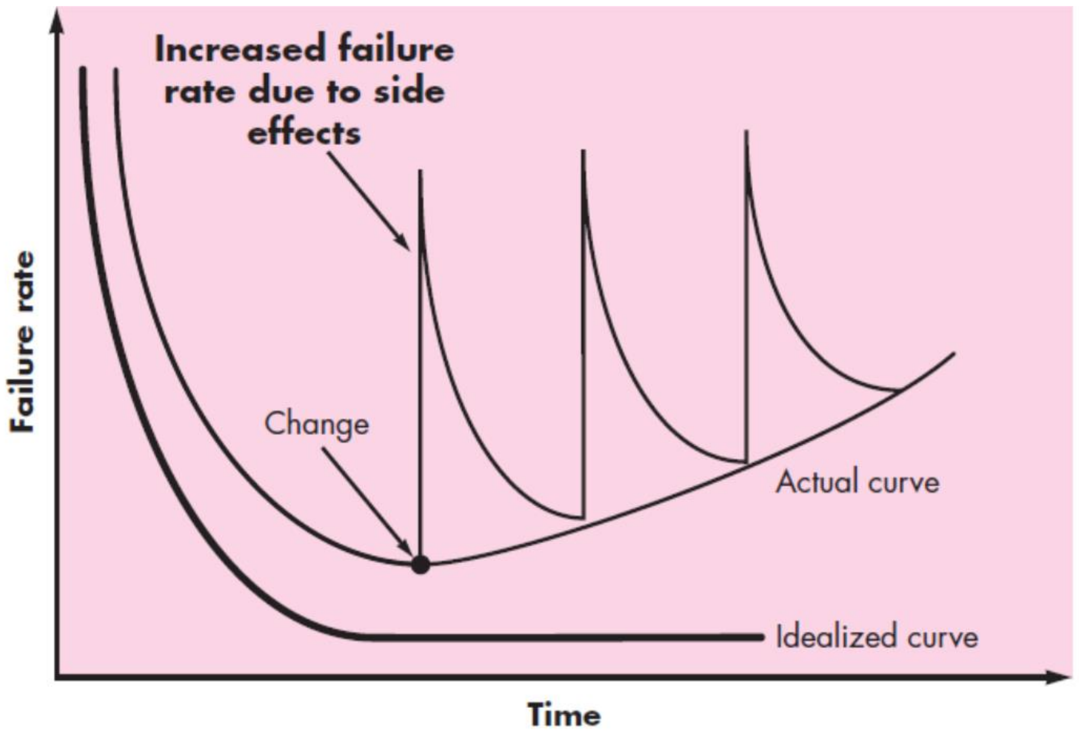
Characteristics of software

- Software is developed or engineered; it is not manufactured in the classical sense.
- Software doesn't "wear out".
  - There is a well-known "bath tub curve" in reliability studies for hardware products. The shape of the curve is like bath tub so known as bath tub curve. This fig shows the failure intensity of hardware product.
  - The three phases in h/w product shows :
    i. The initial phase where failure intensity is higher,
    ii. The expected life phase where failure rate is somewhat stable and low due to extensive testing and failure correction, and
    iii. Finally the wear out phase where failure rate goes up due to the wearing out of the components.



- However in the case of software, software continues to grow instead the software become obsolete due to new operating system environment and new user requirements.
- It can only retire but not wear out.
  - Software does not loose its functionality with use but may loose due to change is user requirements with time and technology.
  - When defects are encountered, they can be removed by rewriting the relevant code, not by replacing with available code. In software engineering the concept of redesign and re-engineering can be used.
  - The defects nature may changes and that may increase the failure rate but these can be minimized and the software life can be rejuvenated.





Characteristics of Software contd…

- Although the industry is moving toward component-based construction, most software continues to be custom built.
- Time and effort for software development are hard to estimate.
- Testing the software is extremely difficult.
- **Attributes of Good Software**
  - Operational: Operational characteristics of software can be measured by
    - Budget,Efficiency,Usability,Dependability,Correctness,Functionality,Safety,Security etc
  - Transitional: the transitional characteristic of software can be measured by
    - Interoperability, Reusability,Portability,Adaptability etc.
  - Maintenable:Maintenable characteristics of software can be measured by
    - Flexibility,Maintainability,Modularity,Scalability etc.

Evolving Role of Software

- Dual Role of software: It is a **product** and, at the same time, the **vehicle for delivering a product**.
  - **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.
  - **As the vehicle** used to deliver the product,
    - It acts as the basis for the control of the computer (operating systems),
    - the communication of information (networks),
    - and the creation and control of other programs (software tools and environments).
    - Software delivers the most important product of our time—information.
- The role of computer software has undergone significant change over the last half-century.
- Sophistication and complexity can produce amazing results when a system succeeds, but they can also pose huge problems for those who must build complex systems.
- Dramatic improvements in hardware performance, profound changes in computing architectures, vast increases in memory and storage capacity, and a wide variety of exotic input and output options have all precipitated more sophisticated and complex computer-based systems.
- Lone programmer has been replaced by a team of software specialists.
- Common questions to lone programmer and group of specialists:
  - Why does it take so long to get software finished?
  - Why are development costs so high?
  - Why can’t we find all errors before we give the software to our customers?
  - Why do we spend so much time and effort maintaining existing programs?
  - Why do we continue to have difficulty in measuring progress as software is being developed and maintained?

Definition of Software Engineering

- The economies of ALL developed nations are dependent on software.
- More and more systems are software controlled
- Software engineering is concerned with theories, methods and tools for professional software development.
- Expenditure on software represents a significant fraction of GNP in all developed countries.
- Software engineering is defined as a process of analyzing user requirements and then designing, building, and testing software application which will satisfy those requirements.
- 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.
- Engineering discipline
  - Using appropriate theories and methods to solve problems bearing in mind organizational and financial constraints.
- All aspects of software production
  - Not just technical process of development. Also project management and the development of tools, methods etc. to support software production.

- **SE Definitions…**
  - “The establishment and use of sound engineering principles in order to obtain economically developed software that is reliable and works efficiently on real machines.” – **Fritz Bauer on first conference on software engineering in 1968**.
  - “A discipline whose aim is the production of quality software, software that is delivered on time, within budget, and that satisfies its requirements.” **– Stephen Schach**.
  - “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.” – **IEEE**
  - “The systematic approach to the development, operation, maintenance, and retirement of software”, where “software” is defined as: “Computer programs, procedures, rules and associated documentation and data pertaining to the operation of a computer system.” – **Boehm**

Importance of Software Engineering

1. **Reduces complexity**
   - Software engineering has a great solution to decrease the complexity of any project. Software engineering divides big problems into several small problems. And then start solving each small problem one by one. All these small problems are solved independently to each other.

2. To minimize software cost
   - In software engineering, programmers plan everything and reduce all those things that are not required. In turn, cost for software productions becomes less as compared to any software that does not use software engineering approach.

3. To decrease time
   - If you are making big software then you may need to run many code to get the ultimate running code. This is a very time consuming process and if it is not well managed then this can take a lot of time. So if you are making your software according to software engineering approach then it will reduce a lot of time.

4. Handling big projects
   - Big projects are not made in few days and they require lots of patience, planning and management. No one can say that he has given four months of company to the project and the program is still in its first stage. Because company has given many resources to the projects and it should be completed. So to handle big projects without any problem, company has to go for software engineering approach.

5. Reliable software
   - Software should be reliable, means if you have delivered the software then it should work for at least it's given time span or subscription. And if any bugs come in the software then company is responsible for solving all these bugs. Because in software engineering, testing and maintenance is provided so there is no worry of its reliability.

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

7. Continuous Production and Innovation

Software costs

- Software costs often dominate computer system costs. The costs of software on a PC are often greater than the hardware cost.
- Software costs more to maintain than it does to develop. For systems with a long life, maintenance costs may be several times development costs.
- Software engineering is concerned with cost-effective software development.
- Roughly 60% of software costs are development costs, 40% are testing costs. For custom software, evolution costs often exceed development costs.
- Costs vary depending on the type of system being developed and the requirements of system attributes such as performance and system reliability
- Distribution of costs depends on the development model that is used
- The cost of software is depending on:
    - direct cost(for example hiring specialists, buying licenses etc.),
    - indirect cost(for example office rent, lighting etc.),
    - fixed costs(for example cost of advertisement etc.),
    - variable costs(staff salary, machine hired etc.)

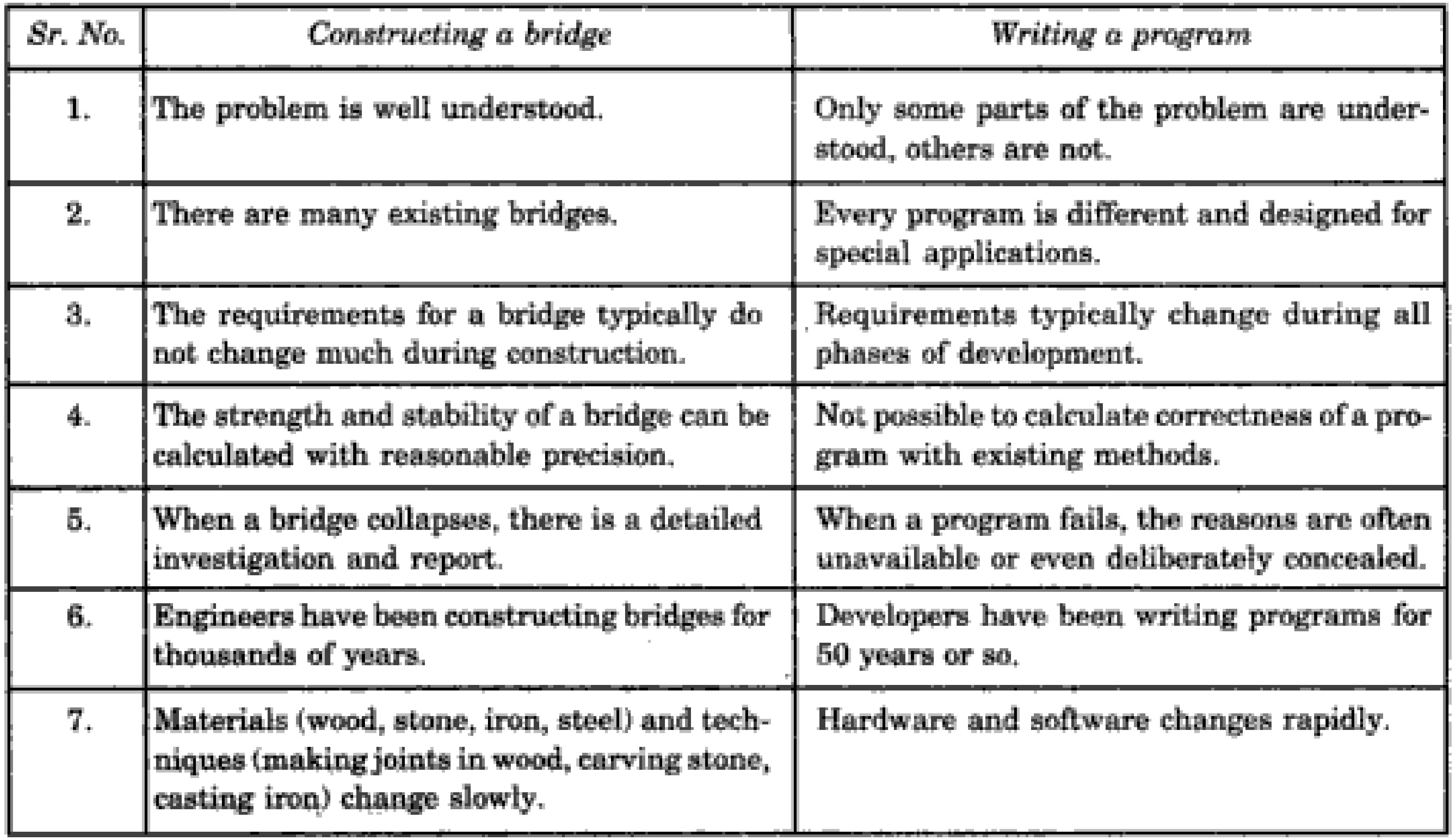
Key Challenges that the Software Engineering Facing

- Coping with legacy systems, coping with increasing diversity and coping with demands for reduced delivery times
- Legacy systems
  - Old, valuable systems must be maintained and updated
- Heterogeneity
  - Systems are distributed and include a mix of hardware and software
- Delivery
  - There is increasing pressure for faster delivery of software

Software Engineering and System Engineering

- **System engineering** is concerned with all aspects of computer-based systems development including hardware, software and process engineering. **Software engineering** is part of this more general process.
- **Software engineering** is concerned with all aspect of the development and evolution of the computer system or other complex system. Whereas software plays major roles in system engineering. **System engineering** is there for, concerned with hardware development process design.
- **Software engineering** highly focuses on implementing quality software while system engineers highly concern about the users and domains.

Example:



Sr. No.	*Constructing a bridge*	*Writing a program*
1.	The problem is well understood.	Only some parts of the problem are understood, others are not.
2.	There are many existing bridges.	Every program is different and designed for special applications.
3.	The requirements for a bridge typically do not change much during construction.	Requirements typically change during all phases of development.
4.	The strength and stability of a bridge can be calculated with reasonable precision.	Not possible to calculate correctness of a program with existing methods.
5.	When a bridge collapses, there is a detailed investigation and report.	When a program fails, the reasons are often unavailable or even deliberately concealed.
6.	Engineers have been constructing bridges for thousands of years.	Developers have been writing programs for 50 years or so.
7.	Materials (wood, stone, iron, steel) and techniques (making joints in wood, carving stone, casting iron) change slowly.	Hardware and software changes rapidly.

Professional and Ethical Practices/Responsibilities

- Software engineering involves wider responsibilities than simply the application of technical skills
- Software engineers must behave in an honest and ethically responsible way if they are to be respected as professionals
- Ethical behavior is more than simply upholding the law.
- Following are the main issues of Professional Practices:
  - **Confidentiality**
    - Engineers should normally respect the confidentiality of their employers or clients irrespective of whether or not a formal confidentiality agreement has been signed.
  - **Competence**
    - Engineers should not misrepresent their level of competence. They should not knowingly accept work which is out with their competence.
  - **Intellectual property rights**
    - Engineers should be aware of local laws governing the use of intellectual property such as patents, copyright, etc. They should be careful to ensure that the intellectual property of employers and clients is protected.
  - **Computer misuse**
    - Software engineers should not use their technical skills to misuse other people’s computers. Computer misuse ranges from relatively trivial (game playing on an employer’s machine, say) to extremely serious (dissemination of viruses).

ACM(Association for computing Machinery)/IEEE Code of Ethics

- The professional societies in the US have cooperated to produce a code of ethical practice.
- Members of these organisations sign up to the code of practice when they join.
- The Code contains **eight Principles** related to the behaviour 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.
  - 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.