Software Engineering

Bsc. CSIT and BCA

Compiled by : T.R. JOSHI

Syllabus

- Unit -1: Introduction
- Unit - 2: Software Development process model
- Unit - 3: Software Requirement Analysis and specification
- Unit - 4: Software Design
- Unit - 5: Coding
- Unit - 6: Software Testing and Quality Assurance
- Unit - 7: Software Maintenance
- Unit - 8: Managing Software Projects

Unit : 1 - Introduction: Contents

- Software and its Types
- Attributes of Good Software
- Software Engineering and its Importance
- Fundamental Software Engineering Activities
- Difference between Software Engineering and Computer Science
- Difference between Software Engineering and System Engineering
- Challenges and Cost of Software Engineering
- Professional Software Development
- Software Engineering Diversity
- Internet Software Engineering
- Software Engineering Ethics

Salute to the history

- The notion of software engineering was first proposed in 1968 at a conference held to discuss what was then called the **software crisis.**
- It became clear that individual approaches to program development did not scale up to large and complex software systems.

- These were unreliable, cost more than expected, and were delivered late. Throughout the 1970s and 1980s, a variety of new software engineering techniques and methods were developed, such as structured programming, information hiding, and object-oriented development.
- Tools and standard notations were developed which are the basis of today's software engineering.

Question	Answer
What is software?	Computer programs and associated documentation. Software products may be developed for a particular customer or may be developed for a general market.
What are the attributes of good software?	Good software should deliver the required functionality and performance to the user and should be maintainable, dependable and usable.
What is software engineering?	Software engineering is an engineering discipline that is concerned with all aspects of software production from initial conception to operation and maintenance.
What are the fundamental software engineering activities?	Software specification, software development, software validation and software evolution.
What is the difference between software engineering and computer science?	Computer science focuses on theory and fundamentals; software engineering is concerned with the practicalities of developing and delivering useful software.
What is the difference between 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.

What are the key challenges facing software engineering?	Coping with increasing diversity, demands for reduced delivery times and developing trustworthy software.
What are the costs of software engineering?	Roughly 60% of software costs are development costs, 40% are testing costs. For custom software, evolution costs often exceed development costs.
What are the best software engineering techniques and methods?	While all software projects have to be professionally managed and developed, different techniques are appropriate for different types of system. For example, games should always be developed using a series of prototypes whereas safety critical control systems require a complete and analyzable specification to be developed. There are no methods and techniques that are good for everything.
What differences has the Internet made to software engineering?	Not only has the Internet led to the development of massive, highly distributed, service-based systems, it has also supported the creation of an "app" industry for mobile devices which has changed the economics of software.

Why software fails?

Software failures are a consequence of two factors:

- **Increasing system complexity:** As new software engineering techniques help us to build larger, more complex systems, the demands change.
  - Systems have to be built and delivered more quickly;
  - larger, even more complex systems are required;
  - And systems have to have new capabilities that were previously thought to be impossible.

New software engineering techniques have to be developed to meet new the challenges of delivering more complex software.

Why software fails?

2. Failure to use software engineering methods:

- It is fairly easy to write computer programs without using software engineering methods and techniques.
- Many companies have drifted into software development as their products and services have evolved. They do not use software engineering methods in their everyday work. Consequently, their software is often more expensive and less reliable than it should be.
- We need better software engineering education and training to address this problem

Types of software product

1. **Generic products:**

- These are stand-alone systems that are produced by a development organization and sold on the open market to any customer who is able to buy them.
- Examples of this type of product include apps for mobile devices, software for PCs such as word processors, drawing packages, and project management tools. This kind of software also includes applications designed for a specific market such as library information systems, accounting systems, or systems for maintaining dental records.

Types of software product

2. Customized (or bespoke) software:

- These are systems that are commissioned by and developed for a particular customer.
- A software contractor designs and implements the software especially for that customer.
- Examples of this type of software include control systems for electronic devices, systems written to support a particular business process, and air traffic control systems.

Attributes Of Good Software

- Acceptability
- Maintainability
- Reliability
- Efficiency
- Scalability
- Portability
- Reusability
- Useability
- Security

Attributes Of Good Software

1. **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.
2. **Dependability and security** - Software dependability includes a range of characteristics including reliability, security, and safety.
    a. Dependable software should not cause physical or economic damage in the event of system failure.
    b. Software has to be secure so that malicious users cannot access or damage the system.

Attributes Of Good Software

3. **Efficiency** - Software should not make wasteful use of system resources such as memory and processor cycles. Efficiency therefore includes responsiveness, processing time, resource utilization, etc.

4. **Maintainability** - Software should be written in such a way that it can evolve to meet the changing needs of customers. This is a critical attribute because software change is an inevitable requirement of a changing business environment.

Software Engineering

- Software engineering is strategy for producing quality software.
- It is the establishment and use of sound engineering principles in order to produce quality softwares.
- Software engineering is an engineering discipline which is concerned with all aspects of software production.
- Software engineers should adopt a systematic and organised approach to their work and use appropriate tools and techniques depending on the problem to be solved

Software Engineering

- Software engineering as a discipline provides us with structured technical means of developing and maintaining software.
- It provides methods to perform the tasks that the making of any software requires, analyzing the requirements, designing the system to meet these requirements, constructing the programs, maintaining the system, etc.
- Software engineering tools are used to support the tasks by automating the tasks or parts of the tasks.

Advantage of Software Engineering

1. Improved quality
2. Improved requirement specification
3. Improved cost and schedule estimates
4. Better use of automated tools and techniques.
5. Better maintenance of delivered software
6. Well defined process
7. Improved reliability
8. Improved productivity
9. Less defects in final processes

Why 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
- Software engineering expenditure represents a significant fraction of GNI(Gross National Income) in all developed countries.

Difference between software engineering and computer science

- Computer science is concerned with theories and methods that underline computer software system
- Software engineering is concerned with the practicalities of developing and delivering useful software
- Some knowledge of computer science is essential for software engineers
- Computer science theories are currently insufficient to act as a complete underpinning for software engineering

Fundamentals of software engineering activities

1. **Software specification:** The functionality of the software and constraints(requirement) on its operation must be defined.
2. **Software development:** The software to meet the specification must be produced.
3. **Software validation:** The software must be validated to ensure that it does what the customer wants.
4. **Software evolution:** The software must evolve to meet changing customer needs.

Difference between Software engineering and system Engineering

1. 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.
2. 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.

Challenges and cost

Challenges of Software Engineering

1. **Coping with increasing diversity**
2. **Demands for delivery in time**
3. **Times and developing trustworthy software**

Cost of Software Engineering

Roughly 60% of software costs are development costs, 40% are testing costs. For custom software, evolution costs often exceed development costs.

Software Engineering Diversity

- **Software engineering includes:**
  - **Practical cost,**
  - **Schedule, and dependability issues,**
  - **As well as the needs of software customers and producers.**
- **The specific methods, tools, and techniques used depend on the:**
  - **Organization developing the software**
  - **The type of software, and**
  - **The people involved in the development process.**
- **There are no universal software engineering methods that are suitable for all systems and all companies. Rather, a diverse set of software engineering methods and tools has evolved over the past 50 years.**

Software Engineering Diversity

Most difficult part of software development is to find the best process for the particular application being developed. There are many different types of application, including:

1. Stand-alone applications:

- **These are application systems that run on a personal computer or apps that run on a mobile device.**
- **They include all necessary functionality and may not need to be connected to a network.**
- **Eg: office applications on a PC, CAD programs, photo manipulation software, travel apps, productivity apps, and so on.**

Software Engineering Diversity

2. Interactive transaction-based applications: These are applications that execute on a remote computer and that are accessed by users from their own computers, phones, or tablets.

- **Obviously, these include web applications such as e-commerce applications where you interact with a remote system to buy goods and services.**
- **This class of application also includes business systems, where a business provides access to its systems through a web browser or special-purpose client program and cloud-based services, such as mail and photo sharing.**

Software Engineering Diversity

3. Embedded control systems: These are **software control systems** that **control** and **manage hardware devices.**

- Examples of embedded systems include the **software** in a **mobile (cell)** phone,
- **software** that controls **antilock braking** in a car, and **software** in a **microwave oven** to control the cooking process.

4. Batch processing systems: These are **business systems** that are designed to **process data in large batches**.

- Examples of batch systems are **periodic billing systems**, such as phone billing systems, and salary payment systems.

Software Engineering Diversity

5. Entertainment systems:

- These are systems for personal use that are intended to entertain the user.
- Most of these systems are games of one kind or another, which may run on special-purpose console hardware.
- The quality of the user interaction offered is the most important distinguishing characteristic of entertainment systems.

6. Systems for modeling and simulation: These are systems that are:

- Developed by scientists and engineers to model physical processes or situations, which include many separate, interacting objects.
- They require high-performance parallel systems for execution.

Software Engineering Diversity

7. Data collection and analysis systems:

- Systems that collect data from their environment and send that data to other systems for processing.
- The software may have to interact with sensors and often is installed in a hostile environment such as inside an engine or in a remote location.
- "Big data" analysis may involve cloud-based systems carrying out statistical analysis and looking for relationships in the collected data.

8. Systems of systems:

- These are systems, used in enterprises and other large organizations, that are composed of a number of other software systems.
- Some of these may be generic software products, such as an ERP(Enterprise resource planning) system.

Professional Software Development

- Lots of people write programs. People in business write spreadsheet programs to simplify their jobs;
- Scientists and engineers write programs to process their experimental data;
- Hobbyists write programs for their own interest and enjoyment.
- However, most software development is a professional activity in which software is developed for business purposes, such as information systems and computer-aided design systems.
- The key distinctions are that professional software is intended for use by someone apart from its developer .
- It is maintained and changed throughout its life.

Professional Software Development

- **It support professional software development rather than individual programming.**
- **It includes techniques that support :**
  - **Program specification**
  - **Design, and evolution,**
- **A professionally developed software system is often more than a single program.**

Internet Software Engineering

- The development of the Internet and the WWW has had a profound effect on all of our lives.
- Initially, the web was primarily a universally accessible information store, and it had little effect on software systems. These systems ran on local computers and were only accessible from within an organization.
- Around 2000, the web started to evolve, and more and more functionality was added to browsers. This meant that web-based systems could be accessed using a web browser from any place.
- This led to the development of a vast range of new system products that delivered innovative services, accessed over the web.

Internet Software Engineering

- Instead of writing software and deploying it on users' PCs, the software was deployed on a web server. This made it much cheaper to change and upgrade the software, as there was no need to install the software on every PC.
- Wherever it has been possible to do so, businesses have moved to web-based interaction with company software systems.

Internet Software Engineering

- The notion of software as a service was proposed early in the 21st century This has now become the standard approach to the delivery of web-based system products such as Google Apps, Microsoft Office 365, and Adobe Creative Suite.
- More and more software runs on remote “clouds” instead of local servers and is accessed over the Internet.
- A computing cloud is a huge number of linked computer systems that is shared by many users.
- Users do not buy software but pay according to how much the software is used or are given free access in return for watching adverts that are displayed on their screen.

Internet Software Engineering

- The advent of the web has led to a dramatic change in the way that business software is organized.
- Before the web, business applications were mostly monolithic, single programs running on single computers or computer clusters.
- Communications were local, within an organization. Now, software is highly distributed, sometimes across the world.
- This change in software organization has had a major effect on software engineering for web-based systems.

Internet software engineering

For example:

1. **Software reuse has become the dominant approach for constructing web-based systems. When building these systems, you think about how you can assemble them from preexisting software components and systems, often bundled together in a framework.**
2. **Web-based systems are always developed and delivered incrementally.**
3. **Software may be implemented using service-oriented software engineering, where the software components are stand-alone web services.**

Internet Software Engineering

- The fundamental ideas of software engineering, discussed in the previous section, apply to web-based software, as they do to other types of software.
- Web-based systems are getting larger and larger, so software engineering techniques that deal with scale and complexity are relevant for these systems.

Software Engineering Ethics

- Like other engineering disciplines, software engineering is carried out within a social and legal framework that limits the freedom of people working in that area.
- As a software engineer, you must accept that your job involves wider responsibilities than simply the application of technical skills.
- You must also behave in an ethical and morally responsible way if you are to be respected as a professional engineer.
- It goes without saying that you should uphold normal standards of honesty and integrity. You should not use your skills and abilities to behave in a dishonest way or in a way that will bring disrepute to the software engineering profession.

Software Engineering Ethics

Some of the Ethics are:

1. **Confidentiality:** You should normally respect the confidentiality of your employers or clients regardless of whether or not a formal confidentiality agreement has been signed.
2. **Competence:** You should not misrepresent your level of competence. You should not knowingly accept work that is outside your competence.
3. **Intellectual property rights:** You should be aware of local laws governing the use of intellectual property such as patents and copyright.
4. Ensure that the intellectual property of employers and clients is protected.

Software Engineering Ethics

4. Computer misuse: You should not use your technical skills to misuse other people's computers. Like dissemination of viruses or other malware.

- **Professional societies and institutions have an important role to play in setting ethical standards. Organizations such as the ACM, the IEEE (Institute of Electrical and Electronic Engineers), and the British Computer Society publish a code of professional conduct or code of ethics.**
- **Members of these organizations undertake to follow that code when they sign up for membership. These codes of conduct are generally concerned with fundamental ethical behavior.**

ACM/IEEE-CS Joint Task Force on Software Engineering Ethics and Professional Practice

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.

Assignment - 1

1. Explain why professional software that is developed for a customer is not simply the programs that have been developed and delivered.
2. What is the most important difference between generic software product development and custom software development? What might this mean in practice for users of generic software products?
3. Briefly discuss why it is usually cheaper in the long run to use software engineering methods and techniques for software systems.
4. Software engineering is not only concerned with issues like system heterogeneity, business and social change, trust, and security, but also with ethical issues affecting the domain. Give some examples of ethical issues that have an impact on the software engineering domain.
5. Explain, with examples, why different application types require specialized software engineering techniques to support their design and development.

Literature Review

- https://www.microsoft.com/en-us/research/uploads/prod/2019/03/amershi-icse-2019_Software_Engineering_for_Machine_Learning.pdf
- Review the paper and write a short abstract on that basis