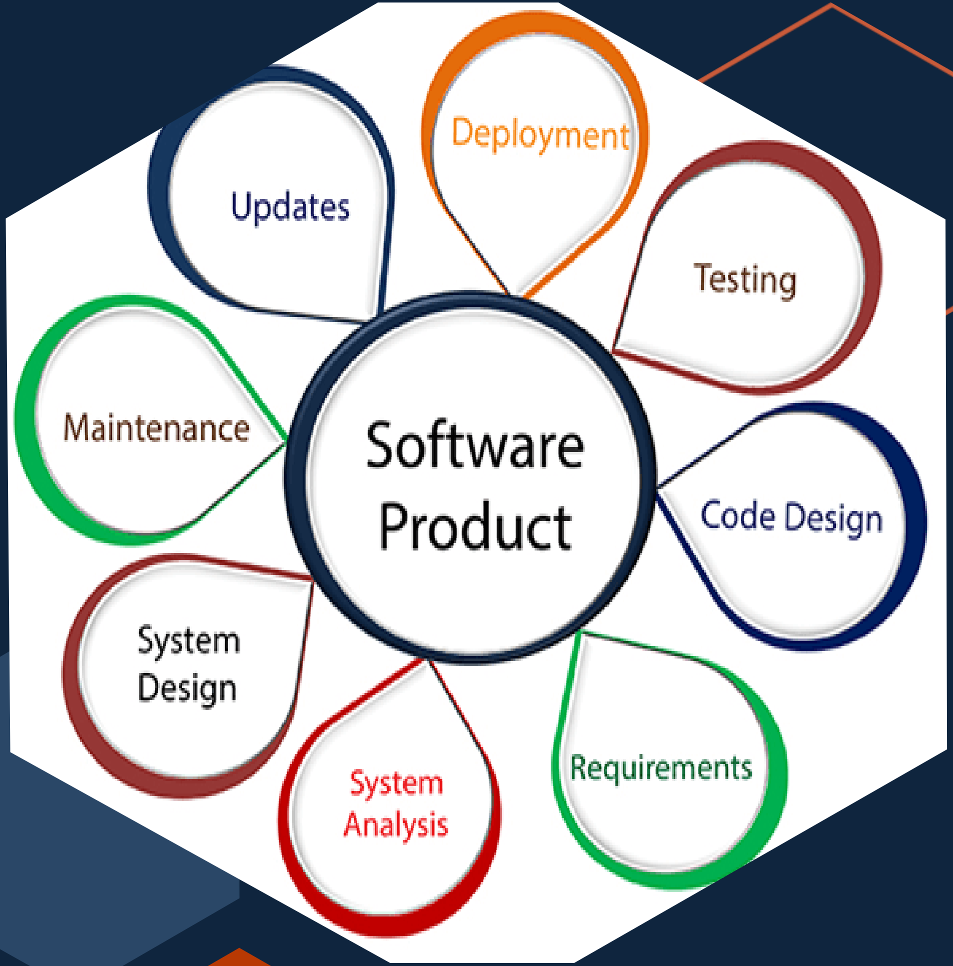
AROR UNIVERSITY
OF ART, ARCHITECTURE,
DESIGN & HERITAGE,
SUKKUR, SINDH

Introduction to Software Engineering

Prof. Dr. Faheem Akhtar Rajput

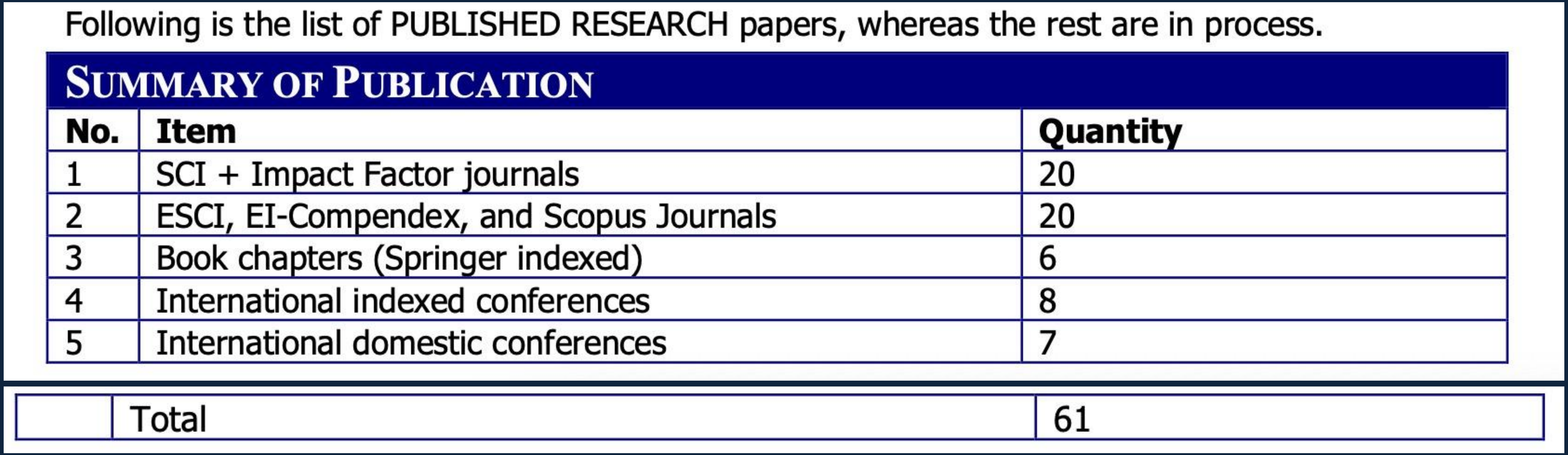


About ME
Prof. Dr. Faheem Akhtar Rajputt

- Professor (Artificial Intelligence & Multimedia Gaming / Cyber Security) at Aror Univeristy
- Associate Professor (Computer Science on Lien) at Sukkur IBA University
- Worked at Bahira University Karachi
- Worked at Muhammad Ali Jinnah University Karachi
- Worked at National University of Computing and Emerging Sciences – FAST Karachi
- Taught in various other universities in KHI as visiting faculty
- Worked in WebXZone Software House

- Won Outstanding International Student Award (Ph.D.) from the Ministry of Education China in 2019.
- Won Outstanding Graduate Award (Ph.D.) from Beijing University of Technology, China in 2020.
- Won Special Contribution Award for outstanding performance in curricular and extracurricular activities (Ph.D.) from Beijing Municipal Government and Beijing University of Technology, China.
- Awarded with a Certificate of Appreciation from the Embassy of Pakistan in Beijing for outstanding Academic performance during my Ph.D. at Beijing University of technology China.
- Received Best Teacher Award in the Computer Science department from the “Aao Parhao campaign” at Sukkur IBA University.

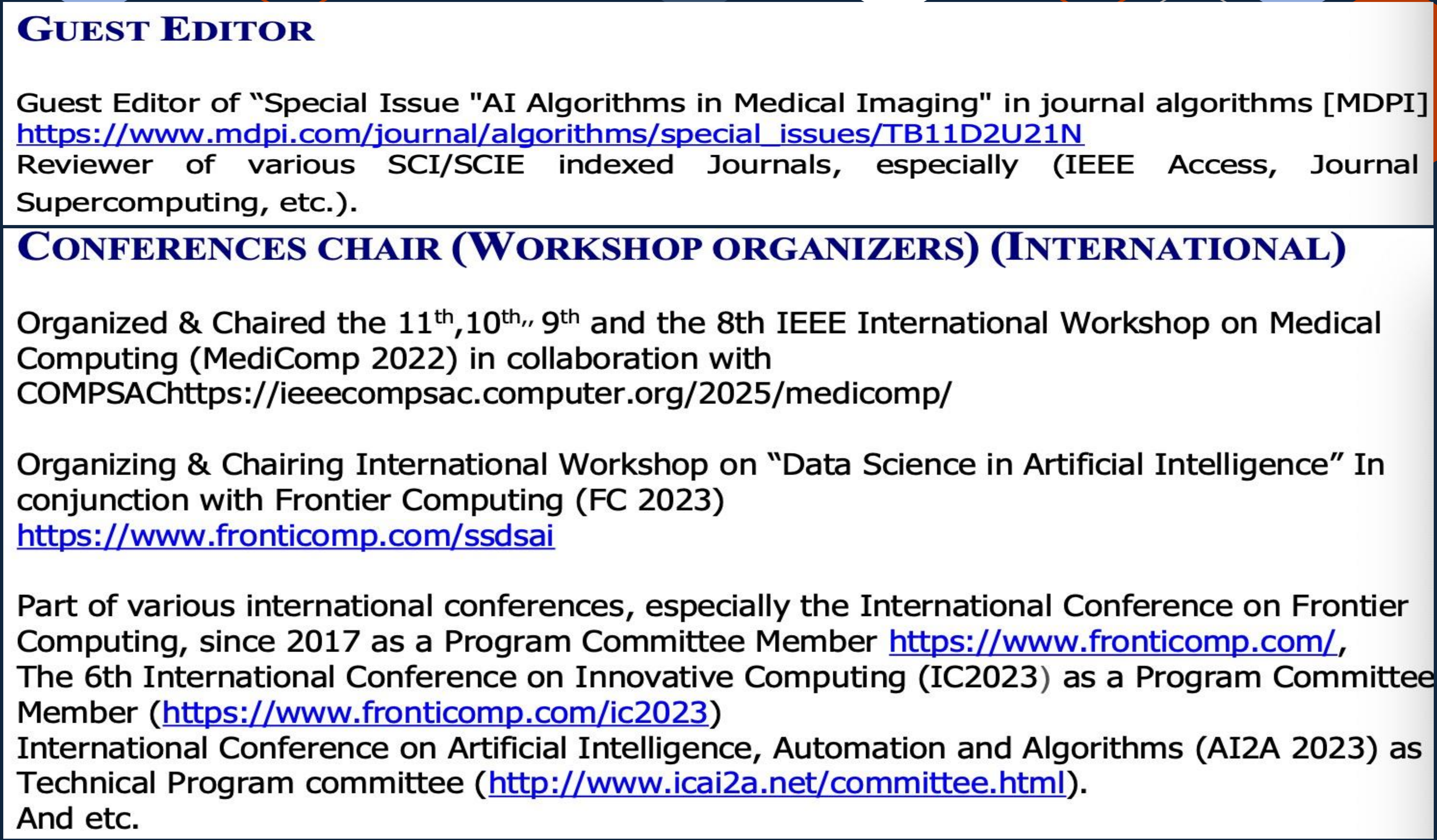
Research Profile



Following is the list of PUBLISHED RESEARCH papers, whereas the rest are in process.

SUMMARY OF PUBLICATION

No.	Item	Quantity
1	SCI + Impact Factor journals	20
2	ESCI, EI-Compendex, and Scopus Journals	20
3	Book chapters (Springer indexed)	6
4	International indexed conferences	8
5	International domestic conferences	7
	Total	61



GUEST EDITOR

Guest Editor of “Special Issue "AI Algorithms in Medical Imaging" in journal algorithms [MDPI] https://www.mdpi.com/journal/algorithms/special_issues/TB11D2U21N
Reviewer of various SCI/SCIE indexed Journals, especially (IEEE Access, Journal Supercomputing, etc.).

CONFERENCES CHAIR (WORKSHOP ORGANIZERS) (INTERNATIONAL)

Organized & Chaired the 11th,10th,, 9th and the 8th IEEE International Workshop on Medical Computing (MediComp 2022) in collaboration with COMPSAChttps://ieeecompsac.computer.org/2025/medicomp/

Organizing & Chairing International Workshop on “Data Science in Artificial Intelligence” In conjunction with Frontier Computing (FC 2023)
https://www.fronticomp.com/ssdsai

Part of various international conferences, especially the International Conference on Frontier Computing, since 2017 as a Program Committee Member https://www.fronticomp.com/,
The 6th International Conference on Innovative Computing (IC2023) as a Program Committee Member (https://www.fronticomp.com/ic2023)
International Conference on Artificial Intelligence, Automation and Algorithms (AI2A 2023) as Technical Program committee (http://www.icai2a.net/committee.html).
And etc.

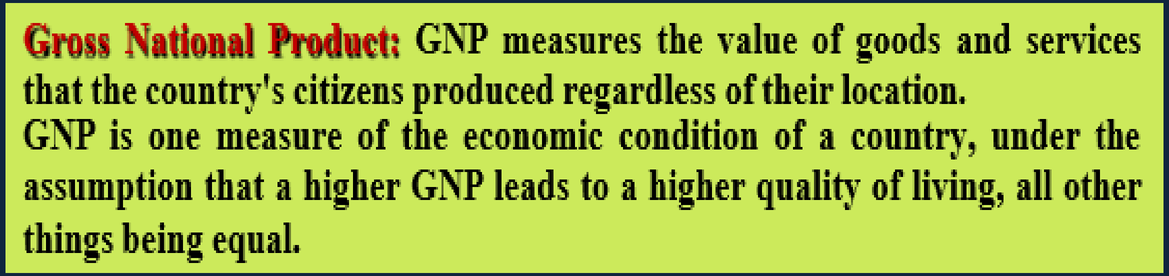
Objectives

- To introduce software engineering and to explain its importance
- To set out the answers to key questions about software engineering
- To introduce ethical and professional issues and to explain why they are of concern to software engineers

Background

- Software Engineering is an engineering discipline whose focus is the cost effective development of high-quality software systems.
- The notation of ***software Engineering*** was first proposed in 1968 at a conference held to discuss what was then called the '***software crisis***'.
- This software crisis resulted directly from the introduction of new computer hardware based on integrated circuits.
- Their power made hitherto unrealisable computer applications a feasible proposition.
- The resulting software was orders of magnitude larger and more complex than previous software systems.

Software Engineering


Gross National Product: GNP measures the value of goods and services that the country's citizens produced regardless of their location.
GNP is one measure of the economic condition of a country, under the assumption that a higher GNP leads to a higher quality of living, all other things being equal.

- 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 GNP in all developed countries

Software Costs

- Software costs often dominate 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 the development costs
- Software engineering is concerned with cost-effective software development

FAQs about Software Engineering

- What is software?
- What is software engineering?
- What is the difference between software engineering and computer science?
- What is the difference between software engineering and system engineering?
- What is a software process?
- What is a software process model?

FAQs about Software Engineering (Cont.)

- What are the costs of software engineering?
- What are software engineering methods?
- What is CASE (Computer-Aided Software Engineering)
- What are the attributes of good software?
- What are the key challenges facing software engineering?

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
- Software products may be
  - **Generic** - developed to be sold to a range of different customers
    - Stand-alone systems that are marketed and sold to any customer who wishes to buy them.
    - Examples – PC software such as graphics programs, project management tools; CAD software; software for specific markets such as appointment systems for dentists.
  - **Bespoke** (custom) - developed for a single customer according to their specification
    - Software that is commissioned by a specific customer to meet their own needs.
    - Examples – embedded control systems, air traffic control software, traffic monitoring systems.

What is Software Engineering?

- 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, the development constraints, and the resources available

What is the difference between software engineering and computer science?

- Computer science is concerned with theory and fundamentals; software engineering is concerned with the practicalities of developing and delivering useful software
- Computer science theories are currently insufficient to act as a complete underpinning for software engineering

What is the difference between software engineering and computer science? (Cont.)

- System engineering is concerned with all aspects of computer-based systems development including hardware, software and process engineering. Software engineering is part of this process
- System engineers are involved in system specification, architectural design, integration and deployment

Why software engineering is important?

- Software engineering is important for two reasons:

1. More and more, individuals and society rely on advanced software systems.
   - We 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 professional software systems rather than just write programs as a personal programming project.
   - Failure to use software engineering method leads to higher costs for testing, quality assurance, and long-term maintenance.

What is software process?

- The systematic approach that is used in software engineering is sometimes called a software process.
- A software process is a sequence of activities that leads to the production of a software product.
- There are four fundamental activities that are common to all software processes.

What is software process?

- Four fundamental activities in software process are:
  - Specification - what the system should do and its development constraints
    - Where customers and engineers define the software that is to be produced and the constraints on its operations
  - Development - production of the software system
    - Where the software is designed and programmed
  - Validation - checking that the software is what the customer wants
    - Where the software is checked to ensure that it is what the customer requires
  - Evolution - changing the software in response to changing demands
    - Where the software is modified to reflect changing customer and market requirements

What is a software process model?

- A simplified representation of a software process, presented from a specific perspective
- Examples of process perspectives are
  - Workflow perspective - sequence of activities
  - Data-flow perspective - information flow
  - Role/action perspective - who does what
- Generic process models
  - Waterfall
  - Incremental development
  - Formal transformation
  - Integration from reusable components

What are the costs of software engineering?

- Roughly 60% of 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
- The distribution of costs depends on the development model that is used

What are the attributes of good software?

- The software should deliver the required functionality and performance to the user and should be maintainable, dependable and usable

1. **Maintainability**

   Software must evolve to meet changing needs

2. **Efficiency**

   Software should not make wasteful use of system resources

3. **Dependability**

   Software must be trustworthy

4. **Usability**

   Software must be usable by the users for which it was designed

Essential Attributes of Good Software

Product characteristic	Description
Maintainability	Software should be written in such a way so 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.
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.
Efficiency	Software should not make wasteful use of system resources such as memory and processor cycles. Efficiency therefore includes responsiveness, processing time, memory utilisation, etc.
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.

What are the key challenges facing software engineering?

- **Heterogeneity**
  - Increasingly, systems are required to operate as distributed systems across networks that include different types of computer and mobile devices.
  - The challenge here is to develop techniques for building dependable software that is flexible enough to cope with this heterogeneity.
- **Business and social change**
  - Business and society are changing incredibly quickly as emerging economies develop and new technologies become available.
  - They need to be able to change their existing software and to rapidly
  - develop new software.
  - They need to evolve so that the time required for software to deliver value to its customers is reduced.

What are the key challenges facing software engineering?

- **Security and trust**
  - As software is intertwined with all aspects of our lives, it is essential
  - that we can trust that software.
  - This is especially true for remote software systems accessed through a web page or web service interface.
  - We have to make sure that malicious users cannot successfully attack our software and that information security is maintained.
- **Scale**
  - Software has to be developed across a very wide range of scales, from very small embedded systems in portable or wearable devices through to Internet-scale, cloud-based systems that serve a global community.

What are the key challenges facing software engineering?

- To address these challenges, we will need new tools and techniques as well as innovative ways of combining and using existing software engineering methods.

Frequently asked questions about 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.
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.

Frequently asked questions about software engineering

Question	Answer
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. You can't, therefore, say that one method is better than another.
What differences has the web made to software engineering?	The web has led to the availability of software services and the possibility of developing highly distributed service-based systems. Web-based systems development has led to important advances in programming languages and software reuse.

Software Engineering Diversity

- There are many different types of software system and there is no universal set of software techniques that is applicable to all of these.
- The software engineering methods and tools used depend on the type of application being developed, the requirements of the customer and the background of the development team.

Application types

- **Stand-alone applications**
  - These are application systems that run on a local computer, such as a PC. They include all necessary functionality and do not need to be connected to a network.
  - Examples of such applications are office applications on a PC, CAD programs, photo manipulation software, travel apps, productivity apps, and so on.

Application types (Cont.)

- **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.

Application types (Cont.)

- **Embedded control systems**
  - These are software control systems that control and manage hardware devices.
  - Numerically, there are probably more embedded systems than any other type of system.
  - 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.

Application types (Cont.)

- **Batch processing systems**
  - These are business systems that are designed to process data in large batches.
  - They process large numbers of individual inputs to create corresponding outputs.
  - Examples of batch systems are periodic billing systems, such as phone billing systems, and salary payment systems.
- **Entertainment systems**
  - These are systems that are primarily for personal use and which are intended to entertain the user.

Application types (Cont.)

- **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.
  - These are often computationally intensive and require high-performance parallel systems for execution.
- **Data collection and analysis systems**
  - Data collection systems are systems that collect data from their environment and send that data to other systems for processing.

Application types (Cont.)

- 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 system.
  - Other systems in the assembly may be specially written for that environment.

Software Engineering Fundamentals

- Some fundamental principles apply to all types of software system, irrespective of the development techniques used:
  - Systems should be developed using a managed and understood development process. Of course, different processes are used for different types of software.
  - Dependability and performance are important for all types of system.
  - Understanding and managing the software specification and requirements (what the software should do) are important.
  - Where appropriate, you should reuse software that has already been developed rather than write new software.

Software Engineering and the Web

- The Web is now a platform for running application and organizations are increasingly developing web- based systems rather than local systems.
- Web services (discussed in Chapter 19) allow application functionality to be accessed over the web.
- Cloud computing is an approach to the provision of computer services where applications run remotely on the 'cloud'.
  - Users do not buy software buy pay according to use.

Web Software Engineering

- Software reuse is the dominant approach for constructing web- based systems.
  - When building these systems, you think about how you can assemble
  - them from pre-existing software components and systems.
- Web-based systems should be developed and delivered incrementally.
  - It is now generally recognized that it is impractical to specify all the
  - requirements for such systems in advance.
- User interfaces are constrained by the capabilities of web browsers.
  - Technologies such as AJAX allow rich interfaces to be created within a web browser but are still difficult to use. Web forms with local scripting are more commonly used.

Web-based Software Engineering

- Web-based systems are complex distributed systems but the fundamental principles of software engineering discussed previously are as applicable to them as they are to any other types of system.

- The fundamental ideas of software engineering, discussed in the previous section, apply to web-based software in the same way that they apply to other types of software system.

Professional and ethical responsibility

- 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 behaviour is more than simply upholding the law but involves following a set of principles that are morally correct.

Issues of professional responsibility

- *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 that is out with their competence.

Issues of professional responsibility (Cont.)

- *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/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.

Code of ethics – preamble

- **Preamble**
  - The short version of the code summarizes aspirations at a high level of the abstraction; the clauses that are included in the full version give examples and details of how these aspirations change the way we act as software engineering professionals. Without the aspirations, the details can become legalistic and tedious; without the details, the aspirations can become high sounding but empty; together, the aspirations and the details form a cohesive code.
  - Software engineers shall commit themselves to making the analysis, specification, design, development, testing and maintenance of software a beneficial and respected profession. In accordance with their commitment to the health, safety and welfare of the public, software engineers shall adhere to the following Eight Principles:

Code of ethics – principles

- 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.

Code of ethics – principles (Cont.)

- 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.

Code of ethics – principles (Cont.)

- 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.

Ethical dilemmas

- Disagreement in principle with the policies of senior management.
- Your employer acts in an unethical way and releases a safety-critical system without finishing the testing of the system.
- Participation in the development of military weapons systems or nuclear systems.

Key Points

- Software engineering is an engineering discipline that is concerned with all aspects of software production.

- Software engineering is concerned with cost- effective software development.

- A set of activities whose goal is the development or evolution of software is known as software process.

Key Points

- Essential software product attributes are maintainability, dependability and security, efficiency and acceptability.
- The high-level activities of specification, development, validation and evolution are part of all software processes.
- The fundamental notions of software engineering are universally applicable to all types of system development.

Key Points

- Software engineering is an engineering discipline that is concerned with all aspects of software production.
- Essential software product attributes are maintainability, dependability and security, efficiency and acceptability.
- The high-level activities of specification, development, validation and evolution are part of all software processes.
- The fundamental notions of software engineering are universally applicable to all types of system development.
- There are many different types of system and each requires appropriate software engineering tools and techniques for their development.
- The fundamental ideas of software engineering are applicable to all types of software system.

GNP: Gross National Product

- Gross National Product: GNP is the total value of all final goods and services produced within a nation in a particular year, plus income earned by its citizens (including income of those located abroad), minus income of non-residents located in that country.
- Basically, GNP measures the value of goods and services that the country's citizens produced regardless of their location.
- GNP is one measure of the economic condition of a country, under the assumption that a higher GNP leads to a higher quality of living, all other things being equal.

Thank you

Prof. Dr. Faheem Akhtar Rajput