

SEMESTER PLAN:

Course Title: “Software Engineering-1”.

Credit hours: 3.

Lecture Duration/ Week: 3 hours.

Semester Duration: 15 Weeks.

Medium Of Instruction: English.

Quiz: 4.

Assignment: 4.

Mid Term: 1.

Project/Presentation: 1.

Final exam: 1.

Lecturer: Khurram Jilani.

Reference Materials

Text Book 1: **Software Engineering 8th edition.**

Authors: Ian Sommerville, Addison Wesley.

ISBN-10: 0321313798

Text Book 2: **Systems Analysis and Design, 7th edition.**

Authors: Gary B. Shelly, Thomas J. Cushman and Harry J. Rosenblatt.

ISBN-10: 1423912225

Text Book 3: **Software Engineering: A Practitioner's Approach 7th edition.**

Authors: Roger S. Pressman, McGraw-Hill

ISBN-10: 0073375977.

- **Other than text books, we will utilize few case studies and reference models in order to understand out of the book real time concepts, involved in failure or success of a software products.**
- **This Course will always welcome any productive suggestions.**

Course Content

- Introduction to Software and Software Engineering.
- Software engineering and other disciplines.
- Software Process.
- Software process models.
- Criteria of selecting process model.
- Software project management.
- Requirement engineering process.
- Distributed System Architecture.
- Rapid Application Development.
- Software Reuse.
- Verification and Validation.
- Software Testing.
- Software Quality Assurance.

Course Objective

- To understand what is software engineering.
- To understand importance of communication and social skills.
- To understand the professional and ethical responsibilities.
- To understand the differences and interlink of other disciplines with software engineering.
- To understand why software engineering so important .
- To understand why process visibilities and modeling is important.
- To understand the importance of project management.
- To understand concepts of testing methodologies and techniques.
- To understand the process of hand over and support of the product.

Session 1: Introduction to Software and Software Engineering

- What is a software.
- Software categories.
- Attributes of good software.
- What is software engineering.
- Aspects of software production.
- Hindrance of software engineering.
- Importance of software engineering.
- Software engineering “a social activity”.
- Ethical and professional responsibilities.
- Key challenges of software engineering.

What is Software.

1. A set of data and instructions.
2. A set of commands which performs desired functions when executed.
3. Executable Source code, running on a device with specific functionality.
4. A set of programs that provide instructions to computers .
5. Product that professionals build and provide support for that product.
6. Computer programs and associated documentation such as requirements, design models and user manuals.

Software Products:

- **Generic products**

Stand-alone systems that are marketed and sold to **any customer** who wishes to buy them.

- **Examples:** MS office, Outlook, Adobe Photoshop, Auto cad etc.

- **Customized products**

Software that is commissioned by **a specific customer** to meet their own needs.

- **Examples:** embedded control systems, air traffic control software, traffic monitoring systems, SAP(CRM) etc.

Software categories:

System software:

- Microsoft Windows (XP, 8 and 10).
- Linux (Mint and Ubuntu).
- Unix (Solaris and Red hat).
- Disk operating system – DOS (Windows 95 and 98)).
- Device Driver Software (Intel HD Graphics Driver).

Software categories-Cont.:

Application Software:

- Opera, internet explorer and Chrome (Web Browsers).
- Documentation and Database software (Microsoft Office and MySQL).
- Audio/video Software (iTunes, VLC and Window media player).
- Gaming Software (World of War craft).
- Graphics and Animation Software(Adobe Photo shop and Flash).

Software categories-Cont.:

Embedded software:

- Key pad control of a microwave oven.
- Digital functions of dashboard display in a car.

Virtualization software.

- VM-Ware.
- Oracle.
- Citrix.

HCI-human Computer Interaction Software.

- Tobii-Eye tracking.
- Gesture Tek.
- X-Plane Flight Simulator.

Difference between System software and Application software:

- The software which **runs the system** is a system software.

If there is no instructions for the hardware to perform specific function then Our PC, Laptop, Tablets or mobiles are just materials .

- The software which **runs on the system** for specific functions is an application software.

If we do not have applications to run on the machine the use of machine remains very limited.

Something about software

- Software is **developed** or **engineered**, it is not **manufactured**.

Because a virtual existence can be developed but can not be manufactured. We can see it but we cant touch it.

- Software doesn't **wear out** but it **deteriorates**.

DVD,CD and USB wear out which carry the software but actual software burned on the device just deteriorates and becomes obsolete.

Attributes of good software.

- Functionality.
- Cost effective.
- User friendly.
- Maintainable.
- Dependable.
- Secure.
- Scalable.

What is software engineering ?

The seminal definition:

- the establishment and use of **sound engineering principles** in order to obtain **economical** software that is **reliable and works efficiently on real machines**.

The IEEE definition:

- The application of a **systematic, disciplined, quantifiable approach** to the **development, operation, and maintenance** of software; that is, the application of engineering to software.

Peter Naur & Brian Randell:

- Software engineering is the **establishment and use of sound engineering principles** in order to **obtain economical software** that is **functional, reliable** and works **efficiently on real machines**.

The Easy Definition:

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

Aspects of Software Production:

- Communication.
- Information Collection.
- Brain Storming /Problem formulation.
- Research/Requirement analysis.
- Planning.
- Designing/Modeling/Prototyping .
- Implementation/Deployment.
- Testing.
- Hand over.
- Maintaining/Monitoring.

Why Software engineering is important ?

Question: Name a industry where software's are not used?

The answer is “**none**”, So this is why software engineering is important. The globe is moving fast to be automated and automation is not possible without software's and software engineering.

- Transportation.
- Medical.
- Telecommunications.
- Military.
- Industrial.
- Entertainment.

Software engineering is center of Business ?

What is common in all of them ?



- Boing manufacturers.
- Stock exchange.
- Amazon.
- OLX.
- Call of duty.
- World war craft.
- Google



Business enterprises.
Services oriented architecture.
Plenty of software's .
Enormous source codes.
Making a lot of money.

SOFTWARE ENGINEERING MANAGEMENT:

Cost Management.

- Software costs often dominate computer system costs.
- Software costs “more to maintain” than develop.
- Cost-effective software development.

Time Management.

- Software Engineering is time constrained.
- Time lines are core of any production.

Risk Management.

- Risk analysis is extremely important.
- Any project is highly vulnerable of unexpected events.

Quality Control.

- Quality control is an integral part of a production.
- Quality control must be executed in spiral process.

Software engineering is a social activity

Conversion of user needs into software, requires communication between users, stake holders and designer then, the designer and developers have to communicate with tools and technology to develop the software.

- What client needs.
- Who will use the application.
- Coordinating with the user (scientists, engineers, banking , medical)
- Other departments working on the project(we work on software development other may work on hardware, networks etc.)

What are the key challenges facing software engineering?

Heterogeneity

Developing techniques for building software that can cope with heterogeneous platforms and execution environments.

Delivery

Developing techniques that lead to faster delivery of functional software.

Trust

Developing techniques that demonstrate that software can be trusted by its users.

Example (Communication/Coordination).

Amazon.com.

- Needed a software upgrade
- Did not went as plan
- 90 minutes outage
- Total damages in lost revenue=**2.8 million dollars**(Customers needed something in those 90 minutes).
- SHOULD have been able to upgrade without taking down the systems.

Examples-(scheduling).

Boing 747- Code scheduling:

- Has four millions lines of codes.
- Greater then accumulative nuts in bolts they have.

Code scheduling:

SLOC= Source line of code.

SE=Software Engineer.

- One source line of code – $SLOC/SE * 60 \text{ minutes} = 60 \text{ lines}$.
- 40 hours a week*60=2400S LOC/SE
- 50 weeks/ year*2400=120,000 SLOC/SE
- Need $2 * 10^6$ SLOC:
- $2 * 10^6 \text{ SLOC} / 1.2 * 10^5 \text{ SLOC} = 17 \text{ SE}$

Ethical and professional responsibilities.

- **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).

Ethical and professional responsibilities.

- **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 outwith their competence.

ACM Association for Computing Machinery /IEEE-CS 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**

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 Eight

Code of ethics – The eight principles

- **PUBLIC**

Software engineers shall act consistently with the public interest.

- **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.

- **PRODUCT**

Software engineers shall ensure that their products and related modifications meet the highest professional standards possible.

Code of ethics - principles

- **JUDGMENT-**

Software engineers shall maintain integrity and independence in their professional judgment.

- **MANAGEMENT**

Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.

- **PROFESSION**

Software engineers shall advance the integrity and reputation of the profession consistent with the public interest.

Code of ethics - principles

- **COLLEAGUES**

Software engineers shall be fair to and supportive of their colleagues.

- **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 a discipline that is concerned with all aspects of software production.
- Software engineering is involved most domains of current technologies.
- Key challenges of software engineering is to maintain heterogeneity, delivery and trust.
- Software products consist of developed programs and associated documentation.
- Essential product attributes are functionalities, maintainability, dependability, efficiency and usability.
- Software engineers have lot of social, ethical and professional responsibilities to the engineering profession and society. They should not simply be concerned only about technical issues.

Acknowledgement

- Thank you all for your time, I hope we all have learned something today.

Next Session

- **Software Engineering and other disciplines.**