# CMSC 128: Introduction to Software Engineering

First Examination Reviewer

## Introduction

### Program

A set of instructions written to perform a certain task. May or may not have an "interface". **Example:** Parse the contents of a text file and format its contents appropriately.

### Application

A software intended to run on a specific environment and perform a task.

### Software

- Composed of many programs
- Can perform a number of tasks
- Documented

## Software Engineering

Software engineering is a term that was introduced by **Friedrich Bauer** in October 1968 NATO Software Engineering conference in Garmisch, Germany. It is the *creative activity of understanding the business problem, coming up wih an idea for solution, and desining the 'blueprints' of the solution.*

Software engineering is the aplication of the disciplined approach for the development and maintenance of computer software. A software engineer must understand the customer's business needs and design software to help meet them. Each customer is unique.

Software engineers must possess:

1. The ability to quickly learn new and diverse disciplines and business processes.
2. The ability to communicate with domain experts, extract an abstract model of the problem, and formulate a solution that makes sense in the context of customer's business.
3. The ability to design a software system that will realize the proposed solution and gracefully evolve with the evolving business needs for many years in the future.

## History of Software Engineering

**John Tukey** first coined software engineering in 1958.

**Alan Turing** first established the theoretical concept of a computer in the 1930s.

**Muhammad Al-Khawarezmi** is a 9th century mathematician who introduced the **concept of algorithm**.

**Ada Lovelace** is the first computer programmer who made an algorithm concrete when she first programmed it.

**Alan Kay** pioneered work on window-based graphical user interface. He is also known for coining the term **object-oriented programming** in 1966.

**Ali Mili** contributed to the **formalization of software fault tolerance**, now a major concern for developing secure software systems.

**Barry Boehm** contributed to the area of software engineering economics and software metrics. He has also introduced the **spiral model** for software development.

**Bill Joy** contributed to the development of the **Unix operating system** and the **Java** programming language.

**Brian Kernighan and Dennis Ritchie** were instrumetal in developing the **Unix operating system** and the **C programming language**.

**C.A.R. Hoare** introduced the concepts of **assertions and program proof of correctness**, designed and analyzed well-known algorithms, and developed communicating sequential processes, a formal language for the specification of concurrent processes.

**David Parnas** introduced the concepts of **information hiding**, software interfaces, and software modularity. He has also contributed to software engineering education and the **ethical responsibilities of a software engineer**.

**Donald Knuth** is known fo the design of many well-known computer algorithms and the use of rigorous mathematical techniques for the formal analysis of the complexity of algorithms.

**Edsger Dijkstra** introduced the concept of **structured programming** and carried out in-depth studies of the problems of concurrency and synchronization needed in complex distributed systems.

**Gamma 4** with Erich Gamma, Richard Helm, Ralph Johnsonn, and John Vlissides introduced the concept of **object-oriented design patters** to facilitate software design reuse.

**Fred Brooks** contributed to the development of the **OS/360 operating system software**. He is also known for introducing mythical man-month in software project management.

**Friedrich Bauer** introduced the use of **stacks** for expression evaluation in programming language compilers, contributed to the development of the **ALGOL** and coined the term *software engineering* in the NATO conference held in Germany in 1968.

**Grace Hopper** contributed extensively to the **first compiler** and the **COBOL programming language** in the 1950s.

**Grady Booch** is known for his method for **object-oriented analysis** and design and his co-development of the **Unified Modeling Language (UML)**.

**John Backus** is well known for the invention of **FORTRAN**, **compiler optimization**, and the **Backus-Naur Form** for the formal description of programming language syntax.

**Michael Fagan** contibuted extensively to the area of **software inspection**.

**Niklaus Wirth** introduced the **Pascal programming language** and other languages, and contributed to the idea of **decomposition** and **stepwise refinement**.

**Ole-Johan Dahl and Kristen Nygaard** introduced the **Simula language** which was the first object-oriented programming language.

**Peter Naur** is known for his contributions to the creation of the **ALGOL programming language** and the introduction of formal syntax description language.

**Tom DeMarco and Edward Yourdon** introduced the structured analysis and design approach for specifying and designing software systems.

**Watts Humphrey** is known for his work on the **capability maturity model** developed by the software engineering institute and the personal software process.

**Winston Royce** contributed the well-known model for the management of large software systems whih was later named the **Waterfall model**.

## Software Development Team

### Project Manager (PM)

- Setup administrative direction for the project
- Lays out the initial scheduling and project goals
- Listens to feedback from end users and Business Analyst
- Setup project meetings and resources
- Determines project scope
- Responsible for documents such as SRS

### Team Leader (TL)

- Leads the developer in creating a module, software or application.
- Coordinates in other team leaders and with the project manager.
- Key point personnel in the managing the pacing of the team in terms of development.
- Suggests and/or decides on the implementation to be used as well as the choice of technology.

### Business Analyst

- Examines the existing or ideal organiation and design of systems including businesses, departments, and organizations.
- May conduct a gap analysis
- Gives advices on areas that might be optimized or improved
- Begins to work out a project strategy

### Developer

- Creates and implements a set of progras needed for the software
- Programs are created based on the agreed architecture and design by the team.
- Creates strategies in the implementation of solutions at different levels of system abstraction and technology used.
- Also known as *Software Developer*, *Programmer*, etc.

### Designer

- Responsible for the look and feel of the application based on the team feedback
- Create mockups for review by the team

### Tester

- Ensures that the application/software behaves properly on a combination of input and scenarios.
- Provides test cases to mimic realistic usage of the software.
- Provides comments for improvements

Sometimes, you would hear some of these terms:

**Full Stack Developer** is knowledgeable in various levels of implementation (i.e. business logic, sever, network, interface).

**UX Designer** is responsible for enhancing user experience and satisfaction through improved usability, aesthetics, accessibility, and performace of a piece of software.

**Subject Matter Expert (SME)** is a person with in-depth knowledge from both a business and IT perspective that when shared with others, significantly enhances performance within the organization.