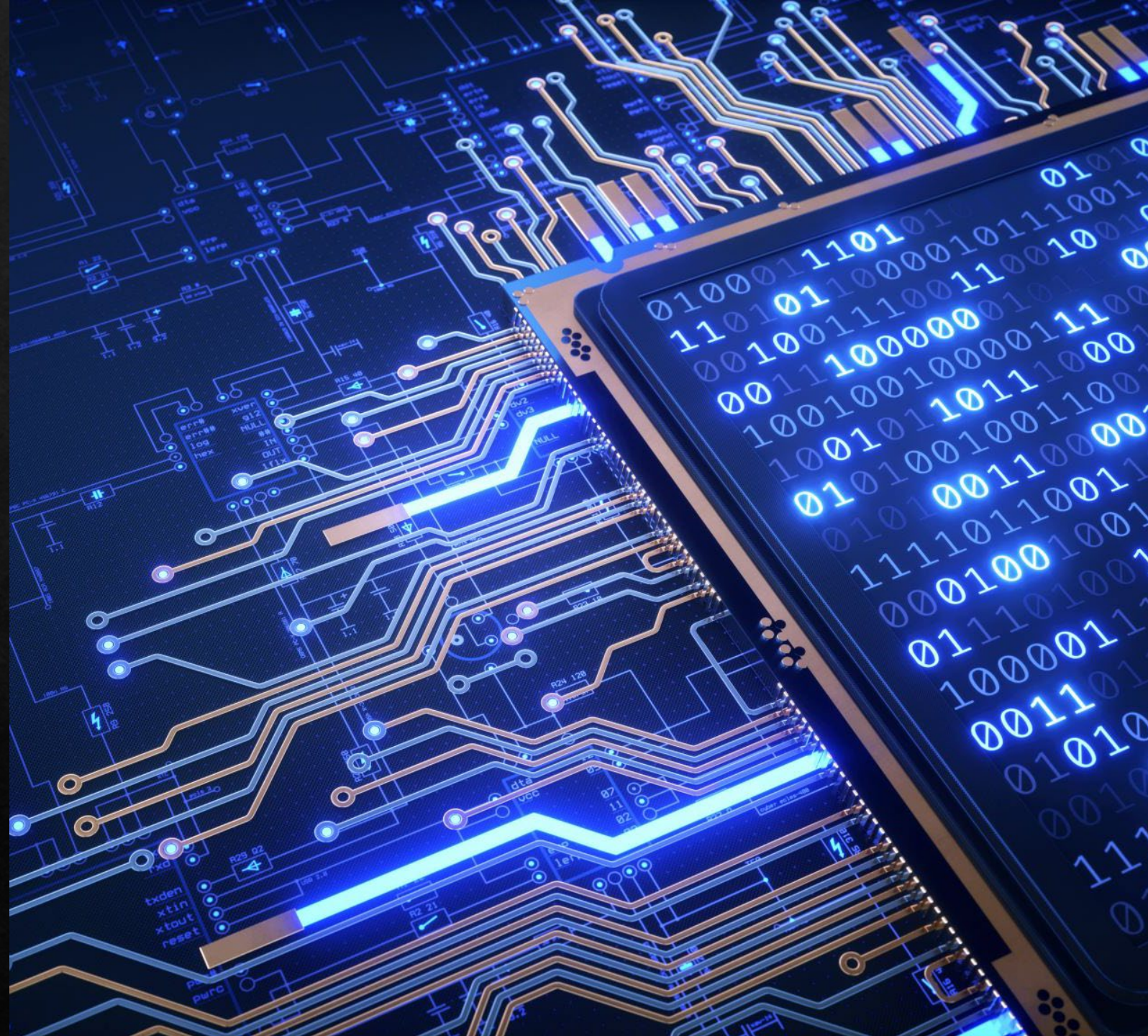


Introduction to Software Engineering

Fethi Rabhi

Acknowledgements: slides prepared by Dr Aarthi
Natarajan



What is software engineering?

- ◇ “The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software, and the study of these approaches; that is, the application of engineering to software.” [IEEE]

What is Software Engineering?

- ◆ **Software Engineering** is a discipline that enables customers to achieve **business goals** through developing software-based systems to solve their **business problems** e.g., develop a course enrolment application or a software to manage inventory.
- ◆ This discipline places great emphasis on the **methodology** or the *method for managing the development process*.
- ◆ The methodology is commonly referred to as **Software Development Life-Cycle (SDLC)**.

Software Engineering is **NOT** Programming

Software engineering:

- **Understanding** the business problem
(understanding the interaction between the system-to-be, its users and environment)
- **Creative formulation** of ideas to solve the problem
 - ◇ based on this understanding
- **Designing** the “blueprint” or architecture of the solution
- Programming:
 - ◇ - **Implementing** the “blueprint” designed by the software engineer

Why become a Software Engineer?

- ◇ Software is **critical to society**

 - ... Software is permeating our society, used to control functions of various machines (e.g., aircrafts, pacemakers)

- ◇ Building software is exciting, challenging and fun

 - ... building software for the next generation that change how billions of people interact, connect and explore

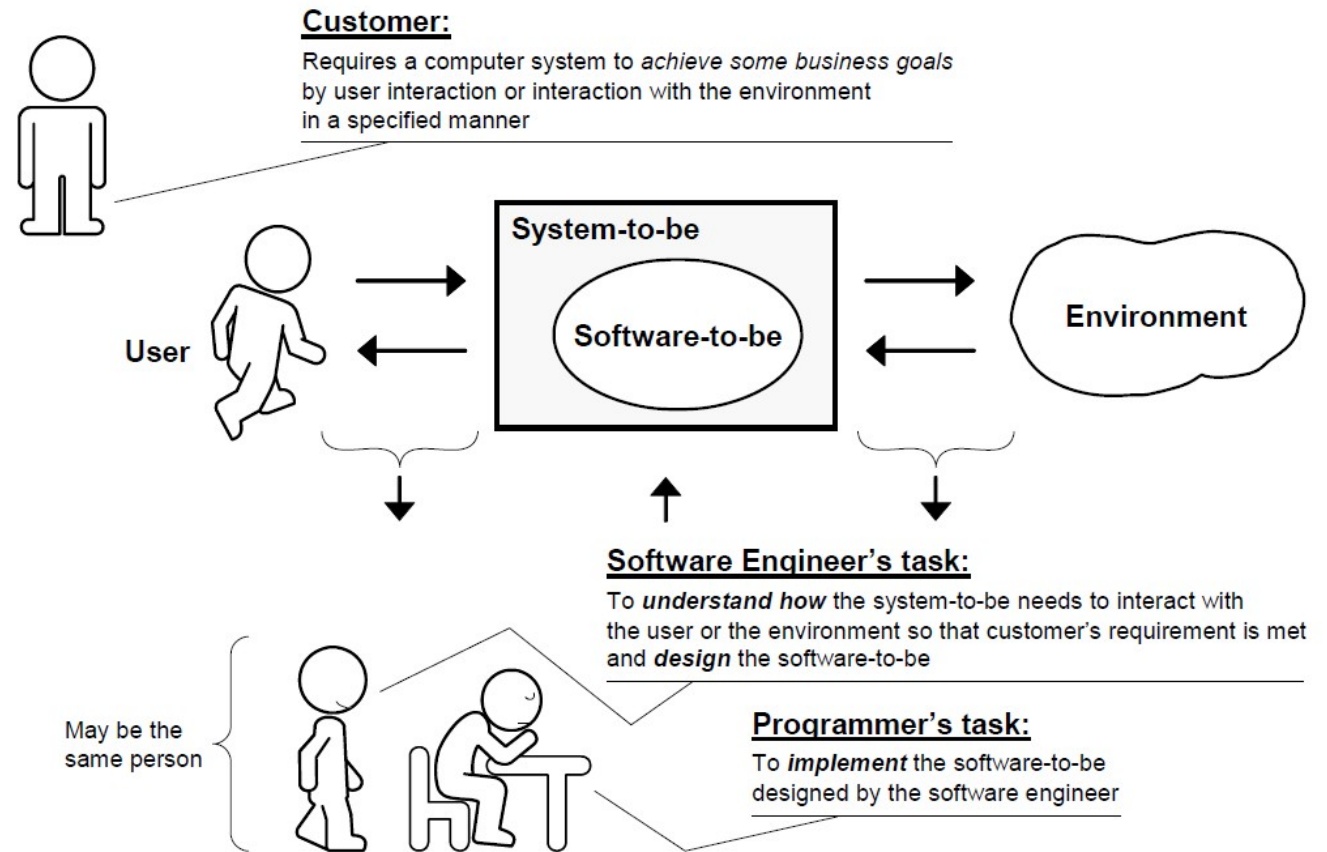
 - ... learn about team culture, meet new people

- ◇ • \$\$\$

Role of a Software Engineer

Software engineer thus acts as a bridge from customer **needs** (problem domain) to programming **implementation** (solution domain).

This enables the software engineer to design solutions that accurately target the customer's needs, that is, deliver **value** to the customer.

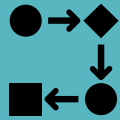


Why do we need SE ?

- ◇ We want to...
 - Make sure what we build is what the customer actually wanted
 - Deliver the software on time and on budget
 - Minimize defects
 - Ensure reliability, security, performance, extensibility, usability, maintainability...
- ◇ To do so, we need a systematic and disciplined approach to software development - and that's what software engineering is about

Difference
between
programmer
& software
engineer





We described software engineering as a complex, organised process with a great emphasis on *methodology*



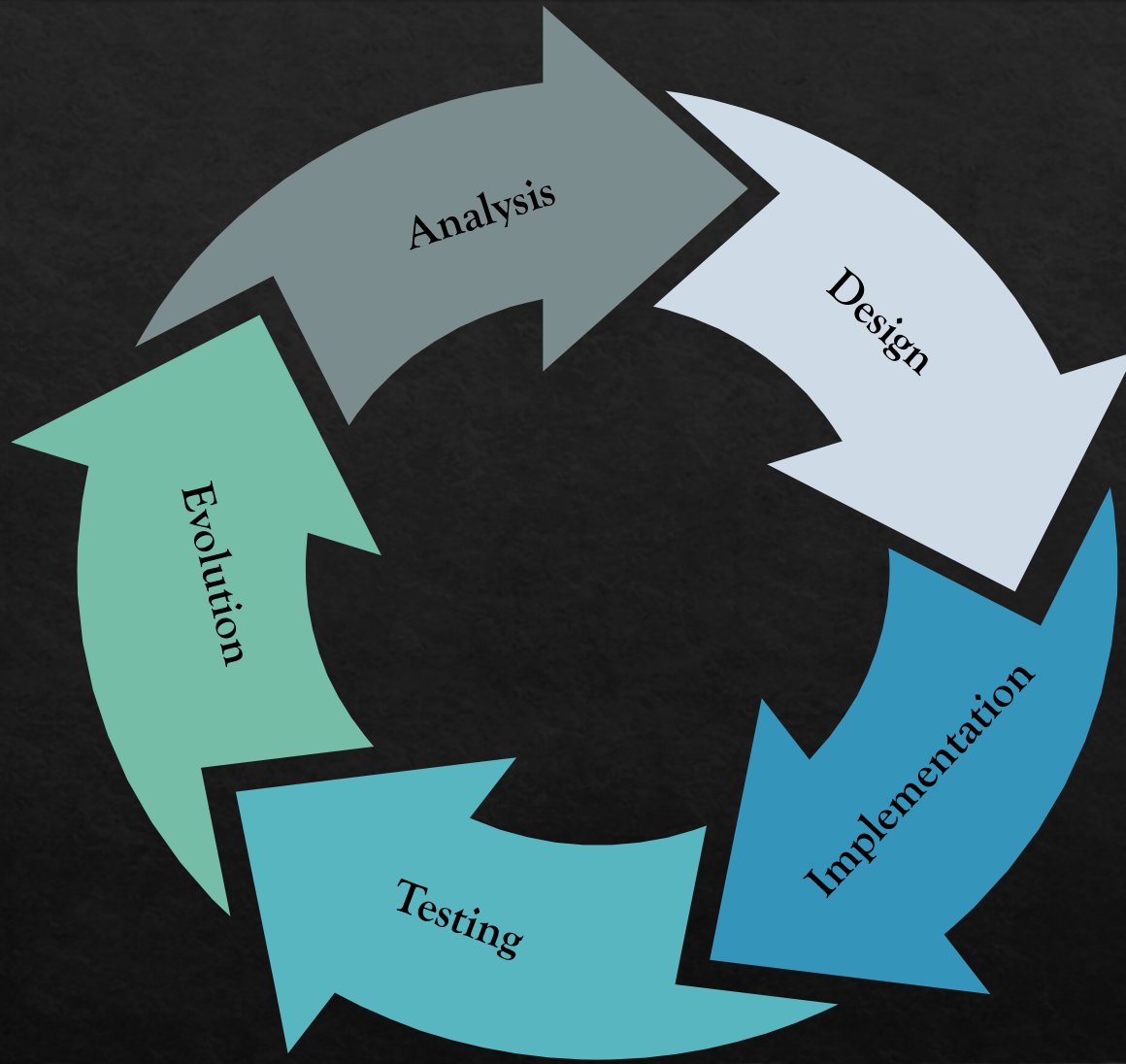
This *methodology* is essentially a framework to structure, plan and control the development of the software system and typically consists of the following phases:

- Analysis and Specification
- Design
- Implementation
- Testing
- Release & Maintenance



Each of the above phases can be accompanied by an artifact or deliverable to be achieved at the completion of this phase

Software Development Life-Cycle



Software Development Life-Cycle

Software Development Challenges

- Software is:
 - probably, the most **complex** artifact
 - **intangible** and hard to visualise
 - the most **flexible** artifact – radically modified at any stage of software development when customer changes requirements
- Waterfall model prescribes a sequential process, but this linear order does not **always** produce best results
- Easier to understand a complex problem by implementing and **evaluating pilot solutions**.

In this course:

- ◆ Practicing SE practices in entrepreneurship
- ◆ Driven by the software product that you are proposing
- ◆ Some guidance given in
 - ◆ Web stacks and Front-End Development
 - ◆ Requirements analysis
 - ◆ User Interface/Experience design

