INTRODUCTION TO SOFTWARE ENGINEERING

LECTURE - 1

SEPTEMBER, 2018

CORE TOPICS TO BE COVERED IN CSD201

- **Software Development Life Cycle Phases**
  - Software Project Planning
  - Software Requirements Engineering
  - Software Design
  - Software Development
  - Software Testing
  - Software Maintenance

SOME GENERAL DEFINITIONS

What is software?

"Computer software or simply software is any set of instructions that directs a computer to perform specific operations." **[wikipedia].**

"Software is considered to be a collection of executable programming code, associated libraries and documentations."

SOME GENERAL DEFINITIONS

What is engineering?

"developing products, using well-defined, scientific principles and methods. "

"Engineering is the application of mathematics, empirical evidence and scientific, economic, social, and practical knowledge in order to invent, innovate, design, build, maintain, research, and improve structures, machines, tools, systems, components, materials, and processes." **[wikipedia]**

What is software engineering?

"Software engineering is the study and an application of engineering to the design, development and maintenance of software" **[wikipedia]**

SOME CONCRETE DEFINITIONS

- **Software**

"instructions (computer programs) that when executed provide desired function and performance, (2) data structures that enable the programs to adequately manipulate information, and (3) documents that describe the operation and use of the programs." **[Pressman]**

- **Software Engineering**

"[Software engineering is] the establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines." **[Pressman]**

SOME CONCRETE DEFINITIONS

Software Engineering

"The application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software" **[IEEE]**

SOFTWARE APPLICATION DOMAINS

- **System software**
- **Application software**
- **Engineering/scientific software**
- **Embedded software**
- **Product-line software**
- **Web Applications**
- **Mobile Applications**
- **Artificial Intelligence Software**

SOFTWARE IS CHANGING

- **Web Apps**
  - Web based systems and applications
- **Mobile Applications**
- **Cloud Computing**
  - Software as a Service (SaaS)
  - Platform as a Service (PaaS)
  - Infrastructure as a Service (IaaS)
- **Product Line Software**

WHY SOFTWARE ENGINEERING IS A TRICKY DISCIPLINE?

- **The ultimate goal is customer satisfaction**
  - However, in general, the customers do not know what they really want
- **Uncertainty exist at every step in each phase of the development life cycle**
- **Requirements keep on changing**
- **Software development is much more than programming**
- **Software has to be maintained after going live**

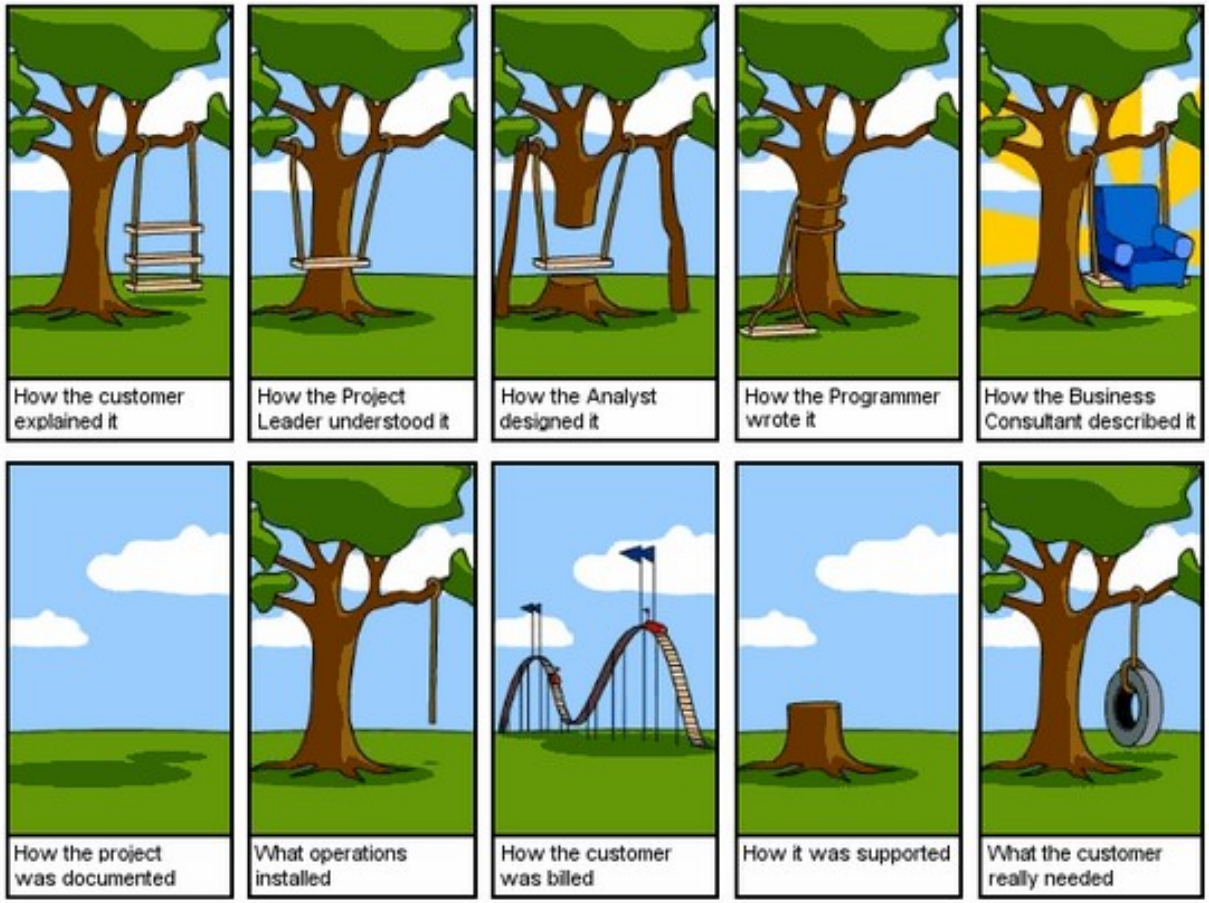
WHY SOFTWARE ENGINEERING IS A TRICKY DISCIPLINE?

- **Up-front detailed estimation and design are complicated yet unnecessary**
- **Have to pay the technical debt**
- **Social activity**
- **…..**

SOFTWARE ENGINEERING FAILURES CASES

- **Faulty Soviet warning system nearly causes WWIII (1983)**
  - The software system malfunctioned and issued a false alarm
  - Lives of millions of human beings could have been at risk
- **The Explosion of the Ariane 5 (1996)**
  - “the self-destruction was triggered by software trying to stuff "a 64-bit number into a 16-bit space”
  - Cost of project failure: 8 billion US dollar
- **Blackout for 50 million across north America (2003)**
  - The reason was a race condition bug in the code

THE DILEMMA


How the customer explained it
How the Project Leader understood it
How the Analyst designed it
How the Programmer wrote it
How the Business Consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed

WHY DOES SOFTWARE FAILURES OCCUR?

- **Lack of management**
- **Lack of clear specifications**
- **Misunderstanding of specifications**
- **Infeasible specification**
- **Coding and transcription errors**
- **Equipment failures**
- **Operating system, compiler and library failures**
- **Changes in external systems**
- **Changes in internal subsystems**
- **Unintended consequences of other changes**
- **Careless or malicious human intervention**

HOW DOES SOFTWARE FAILURES OCCUR?

- **Incorrect code**
- **Missing code**
- **Extra code**
- **Incorrect references to external processes**
- **Code that does not meet time constraints**
- **Code that does not meet space constraints**
- **Human interfaces that dont match people's training or preferences**

HOW TO IMPROVE?

<Class participation>

<Oration by instructor>

Q&A