CS20202: Software Engineering

Module M01: Course Information & Introduction to Software Engineering

Partha Pratim Das

Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

ppd@cse.iitkgp.ac.in

January 05, 2022

Table of Contents

Why Software Engineering?

Engineering: Skills of Construction

- Civil Engineering
  - Construction of Buildings
- Mechanical Engineering
  - Construction of Automobiles
- Electrical Engineering
  - Construction of Power Plants
- **Software Engineering**
  - *Development of Software*

What Software Engineering is NOT!

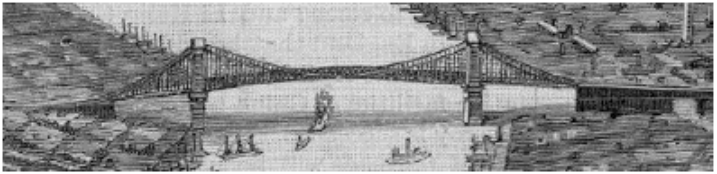
- Programming
- Data Structures
- Algorithms
- Design
- Testing
- Deployment
- Maintenance
- . . .
- . . .
- . . .
- Construction!

Evolution of Domains

- Construction
- Medicine
- Aviation
- Computing
- Software

Construction

- Fallen trees, Stepping stone (10000 BC), Boardwalk
- Arch bridge, 1300 BC
- Iron bridge, 1779
- Concrete Bridges, 1877
- Steel bridge, 1912
- Bailey bridge, 1940
- Constructing a bridge is different from innovating a bridge (with new material for instance) for the first time
- Engineers use well established metrics to design bridges – they do not innovate at this stage



Medicine

- *Health was thought to be restored by purging, starving, vomiting, or bloodletting*
  - Surgeons and barbers specialized in this practice
  - Widely practiced in 18^{th} & 19^{th} century
  - Declared quackery by 1900

- Infection control
  - Survived surgery, died out of infection
  - Germ theory and sterility came only in late 1800s (Lister)
  - Current rate of infection $< 2.5\%$

Aviation

- 400 BC Chinese fly kite aspiring humans to fly
- For centuries, we try to fly like birds... disastrous
- Steam powered, hot air
- Gliders, single man
- Engine powered
- 1903 Wright brothers' first flight – 12s, 120' long, 10' high
- UK's Frank Whittle registered patent for the turbo-jet engine in 1930, first flight test in 1941
- Concorde, 1976, mach 2

Computing

- 1801: In France, Joseph Marie Jacquard made Punch cards for fabric design in **Jacquard Loom**
- 1822: English mathematician **Charles Babbage Machine** (failed)
- 1890: Herman **Hollerith punch card** for 1880 census saves $5 m. His company becomes **IBM**
- 1936: Alan Turing - **Turing machine**
- 1939: **Hewlett-Packard** is founded
- 1941: Atanasoff & Clifford Berry introduces **main memory** to solve 29 equations simultaneously
- 1943-1944: John Mauchly & J. Presper Eckert, build **ENIAC**: 20' X 40', 18,000 vacuum tubes.
- 1946: Mauchly & Presper build the **UNIVAC**, the first commercial computer for business applications
- 1947: William Shockley, John Bardeen and Walter Brattain of Bell Laboratories invent the **transistor**
- 1953: Grace Hopper develops the first computer language **COBOL**
- 1954: The **FORTRAN** programming language, developed by an IBM team led by John Backus
- 1958: Jack Kilby (Physics Nobel, 2000) & Robert Noyce unveil the **integrated circuit**
- 1964: Douglas Engelbart shows a prototype of the modern computer, with **a mouse and a GUI**
- 1969: A group of developers at Bell Labs produce **UNIX**
- 1970: Intel unveils the Intel 1103, the first **Dynamic Access Memory (DRAM)** chip.
- 1971: Alan Shugart leads a team of IBM engineers who invent the **floppy disk**
- 1973: Robert Metcalfe, Xerox, develops **Ethernet**
- 1974-1977: A number of **personal computers** hit the market
- 1975: Paul Allen & Bill Gates, write software for the Altair 8080, using BASIC language; form **Microsoft**
- 1976: Steve Jobs and Steve Wozniak start **Apple Computers** on April Fool's Day
- 1977: Apple offers **color graphics** and incorporates an **audio cassette drive for storage**
- 1978: Accountants **VisiCalc**, the first computerized **spreadsheet program**
- 1979: MicroPro International releases **WordStar**

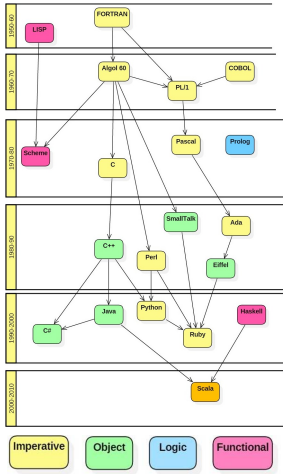
Computing

- 1981: The **first IBM personal computer**, code-named "Acorn," uses Microsoft's **MS-DOS**
- 1983: Apple's **Lisa is the first personal computer with a GUI**
- 1985: Microsoft announces **Windows**
- 1985: First dot-com domain name, **Symbolics.com**, registered on Mar. 15 by Symbolics Computer Company
- 1986: Compaq brings the **Deskpro 386, 32-bit architecture**, providing speed comparable to mainframes
- 1990: Tim Berners-Lee, a researcher at CERN, develops **HTML**, giving rise to the **World Wide Web**
- 1993: The **Pentium microprocessor** advances the use of graphics and music on PCs.
- 1996: Sergey Brin and Larry Page develop the **Google Search Engine** at Stanford University.
- 1997: **Microsoft invests $150 million in Apple**
- 1999: The term **Wi-Fi** becomes part of the computing language
- 2001: Apple unveils the **Mac OS X**
- 2003: **The first 64-bit processor, AMD's Athlon 64**
- 2004: **Mozilla's Firefox 1.0** challenges Microsoft's Internet Explorer. **Facebook** launches
- 2005: **YouTube** is founded. Google acquires **Android**, a Linux-based mobile phone operating system
- 2006: Apple introduces the **MacBook Pro Nintendo's Wii game console** hits the market
- 2007: The **iPhone** brings many computer functions to the smartphone.
- 2010: Apple unveils the iPad
- 2012: **Facebook gains 1 billion users on October 4**
- 2015: Apple releases the **Apple Watch**. Microsoft releases **Windows 10**
- 2016: The first **reprogrammable quantum computer** was created
- 2017: DARPA is developing a new **Molecular Informatics** program that uses molecules as computers.
- 2019: **Corona hits!**

Source: History of Computers: A Brief Timeline

History of Programming Languages

History of Programming Languages



Paradigms: *Imperative*: Algorithms + Data, *Object*: Data, *Logic*: Facts + Rules + Queries, and *Functional*: Functions

- **FORTRAN**: IBM
- **LISP**: John McCarthy
- **Algol 60**: John Backus & Peter Naur
- **COBOL**: Grace Murray Hopper
- **PASCAL**: Niklaus Emil Wirth
- **Prolog**: Alain Colmerauer & Philippe Roussel
- **Scheme**: Guy L. Steele & Gerald Jay Sussman
- **C**: Brian W. Kernighan & Dennis M. Ritchie
- **SmallTalk**: Alan Kay, Dan Ingalls, & Adele Goldberg
- **Ada**: Jean Ichbiah & Tucker Taft
- **C++**: Bjarne Stroustrup
- **Objective-C**: Brad Cox
- **Perl**: Larry Wall
- **Java**: James Gosling
- **Python**: Guido van Rossum
- **Haskell**: Paul Hudak
- **C#**: Microsoft Corporation
- **Ruby**: Yukihiro Matsumoto
- **Scala**: Martin Odersky

Source: Programming Language Evolution

TIOBE Index of Programming Languages

Jan 2021	Jan 2020	Change	Programming Language	Ratings	Change
1	2	^	C	17.38%	+1.61%
2	1	v	Java	11.96%	-4.93%
3	3		Python	11.72%	+2.01%
4	4		C++	7.56%	+1.99%
5	5		C#	3.95%	-1.40%
6	6		Visual Basic	3.84%	-1.44%
7	7		JavaScript	2.20%	-0.25%
8	8		PHP	1.99%	-0.41%
9	18	^^	R	1.90%	+1.10%
10	23	^^	Groovy	1.84%	+1.23%
11	15	^^	Assembly language	1.64%	+0.76%
12	10	v	SQL	1.61%	+0.10%
13	9	vv	Swift	1.43%	-0.36%
14	14		Go	1.41%	+0.51%
15	11	vv	Ruby	1.30%	+0.24%
16	**20**	^^	**MATLAB**	**1.15%**	**+0.41%**
17	19	^	Perl	1.02%	+0.27%
18	13	vv	Objective-C	1.00%	+0.07%
19	12	vv	Delphi/Object Pascal	0.79%	-0.20%
20	16	vv	Classic Visual Basic	0.79%	-0.04%

Jan 2022	Jan 2021	Change	Programming Language	Ratings	Change
1	3	^	Python	13.58%	+1.66%
2	1	v	C	12.44%	-4.94%
3	2	v	Java	10.66%	-1.30%
4	4		C++	8.29%	+0.73%
5	5		C#	5.68%	+1.73%
6	6		Visual Basic	4.74%	+0.90%
7	7		JavaScript	2.09%	-0.11%
8	11	^	Assembly language	1.85%	+0.21%
9	12	^	SQL	1.80%	+0.19%
10	13	^	Swift	1.41%	-0.02%
11	8	v	PHP	1.40%	-0.60%
12	9	v	R	1.25%	-0.65%
13	14	^	Go	1.04%	-0.37%
14	19	^^	Delphi/Object Pascal	0.99%	+0.20%
15	20	^^	Classic Visual Basic	0.98%	+0.19%
16	16		MATLAB	0.96%	-0.19%
17	10	vv	Groovy	0.94%	-0.90%
18	15	v	Ruby	0.88%	-0.43%
19	30	^^	Fortran	0.77%	+0.31%
20	17	v	Perl	0.71%	-0.31%

Source: TIOBE Index of Programming Languages

Software

- Relatively nascent field in comparison
- Machines are getting faster or more powerful
- Are we getting better in delivering software applications though?

Success (or Lack thereof)

- How successful are we in developing software?
- Less than 10% of software projects succeed!
- Criteria for success?
  - On time,
  - Within budget,
  - Feature complete,
  - Works (failure free)
- Why is it so hard to get this right?

Change In Projects

- Changes From Requirements
  - Customers Learn from the Solution
  - Business Environment and Conditions Change
  - Business Processes are Re-engineered
- Changes From Technology
  - Tools/Platform Release New Versions
  - Actual Tool/Platform Capabilities May Vary from Plans
- Changes From People
  - Interactions are Complex
  - Individual Behavior is Unpredictable

Software Engineering

- What's Engineering?
  - *the application of science and mathematics by which the properties of matter and the sources of energy in nature are made useful to people*
  - *the design and manufacture of complex products* <**software engineering**>

Software Engineering

- If software engineering like manufacturing or designing a manufacturing plant?
  - Is it like making another cell phone or making of cell phones (took 37 years for commercialization)?
- Manufacturing is predictive
  - You can measure and control quality, quantity
- Designing a manufacturing plant is creative/innovative
- Most software development is innovative process rather than predictive manufacturing
  - Requires great deal of innovation, interaction / communication

Course: Software Engineering

- We demystify software construction and learn the *good practices*

Agenda: Software Engineering

- Software as a product
  - Clients and their needs
  - Quality
- Requirements and specification
  - Usability
  - Evolution
- Software design
  - Software architecture
  - Object-oriented design

Agenda: Software Engineering

- Software Processes
  - Coding
  - Reading
  - Review
  - Source Management
  - Debugging
  - Testing
  - Reliability
  - Verification
  - Documentation
  - Quality
  - Maintenance

Agenda: Software Engineering

- Project management
  - Personnel management
  - Economic, legal, and social factors
- Standards

Prerequisites

Prerequisites

[1] Programming

[2] Data Structure

[3] Algorithms

[4] Object-Oriented Analysis and Design (*optional*)

Syllabus

Modules

- Module 01: Course Information & Introduction to Software Engineering
- Module 02: Object Oriented Analysis, Design, and Programming (in C++)
- Module 03: Software Development Life Cycle (SDLC) Phase and Models
- Module 04: Software Engineering Processes
- Module 05: Software Quality and Reliability
- Module 06: Software Project Management
- Module 07: Smart Software Engineering

Refer: Syllabus of Software Engineering

Course Information

Course Material

- Slides will be uploaded to Moodle.
- Books:
  - Software Engineering by Rajib Mall
  - Software Engineering: A Practitioner's Approach by Roger S Pressman
  - An Integrated Approach to Software Engineering by Pankaj Jalote
  - Software Project Management – A Process-Driven Approach by Ashfaque Ahmed
  - The Java Programming Language by Ken Arnold, James Gosling, & David Holmes
  - The C++ Programming Language by Bjarne Stroustrup
  - Modern C++ Design by Andrei Alexandrescu
  - Design Patterns: Elements of Reusable Object-Oriented Software by Erich Gamma, Richard Helm, Ralph Johnson, & John Vlissides
  - Learning UML 2.0 – A Pragmatic Introduction to UML by Russ Miles & Kim Hamilton (O'Reilly)
  - Effective C++ & More Effective C++ by Scott Meyers
  - Exceptional C++ & More Exceptional C++ by Herb Sutter

About the Course: Interactions

- Timings: WED(12:00-12:55), THR(11:00-11:55), FRI(9:00-9:55)
- Classes and interactions will be held on Microsoft Teams: Software Engineering CS20202
- Kindly keep your microphone muted
- Kindly keep your video off
- Kindly put your comments / doubts on the chat – chats will be periodically checked and responded
- Kindly raise your hand to ask a question
- Deeper interactions / feedback will be over Forum on Moodle
- Interaction Outside Class: By appointment through mail - over audio / video chat

About the Course: Evaluations

- Assignments
  - In-Class Assignments
    - Marks: 5~10 each. Time: 15-30 minutes. Completion within the class.
  - Offline Assignments
    - Marks: 10~20 each. Time: 1-2 weeks.
  - Total Marks: 70. Total of the assignments will be scaled to 70
  - To be hand-written, scanned and uploaded - write clearly using bigger font styles
- Online Test
  - Marks: 15. Time: 1 hour. # of Test: 3
  - Total Marks: 30. Total of the assignments will be scaled to 30
- Relative Grading
  - Marks of assignments and tests will be added to get to total out of 100
  - Grade boundary will be decided relatively based on the bell curve

The Coordinating Platforms

- Moodle will be used for the course. Register on Moodle immediately to:
  - CS20202 - Software Engineering 2022. Course Key: SE22STU
  - CS29202 - Software Engineering Laboratory 2022. Course Key: SEL22STU
- All assignments / presentations / material will be uploaded to Moodle
- The submissions will be accepted *only* through Moodle up to the specified deadline. No submission through mail will be entertained
- Extensions permissible only on medical ground (B C Roy certificate) and IIT duty (like inter-IIT Sports meet on Dean's Order)
- 10% to 50% penalty (depending on assignment and amount of delay) on late submission on discretionary basis
- Zero tolerance to plagiarized submissions. Penalty applies to both parties
- Online Quiz will be held online in Moodle
- All announcements will be made on Moodle. Keep checking
- ERP will also be used at times for communication. Make sure that your registered email at ERP works
- Recording of class lectures will be posted on YouTube
- Website: https://cse.iitkgp.ac.in/~sourangshu/coursefiles/cs20202_2022s.html

Schedule for Tests & Quizzes

Test / Quiz	Date	Time
Test 1 / Quiz 1	02-Feb-22	2:00–5:00
Test 2 / Quiz 2	23-Feb-22	2:00–5:00
Test 3 / Quiz 3	13-Apr-22	2:00–5:00