Introductory Sessions

Jan 2023

Welcome!

Course CS3004D Software Engineering

Instructors VP/MK/JCR

CR01 / CR02 / CR03 / CR04

Slots - A1+ A2+

Slides, Course Plan, Marks, Announcements
Eduserver

Course Plan

Disclaimer on Dependencies

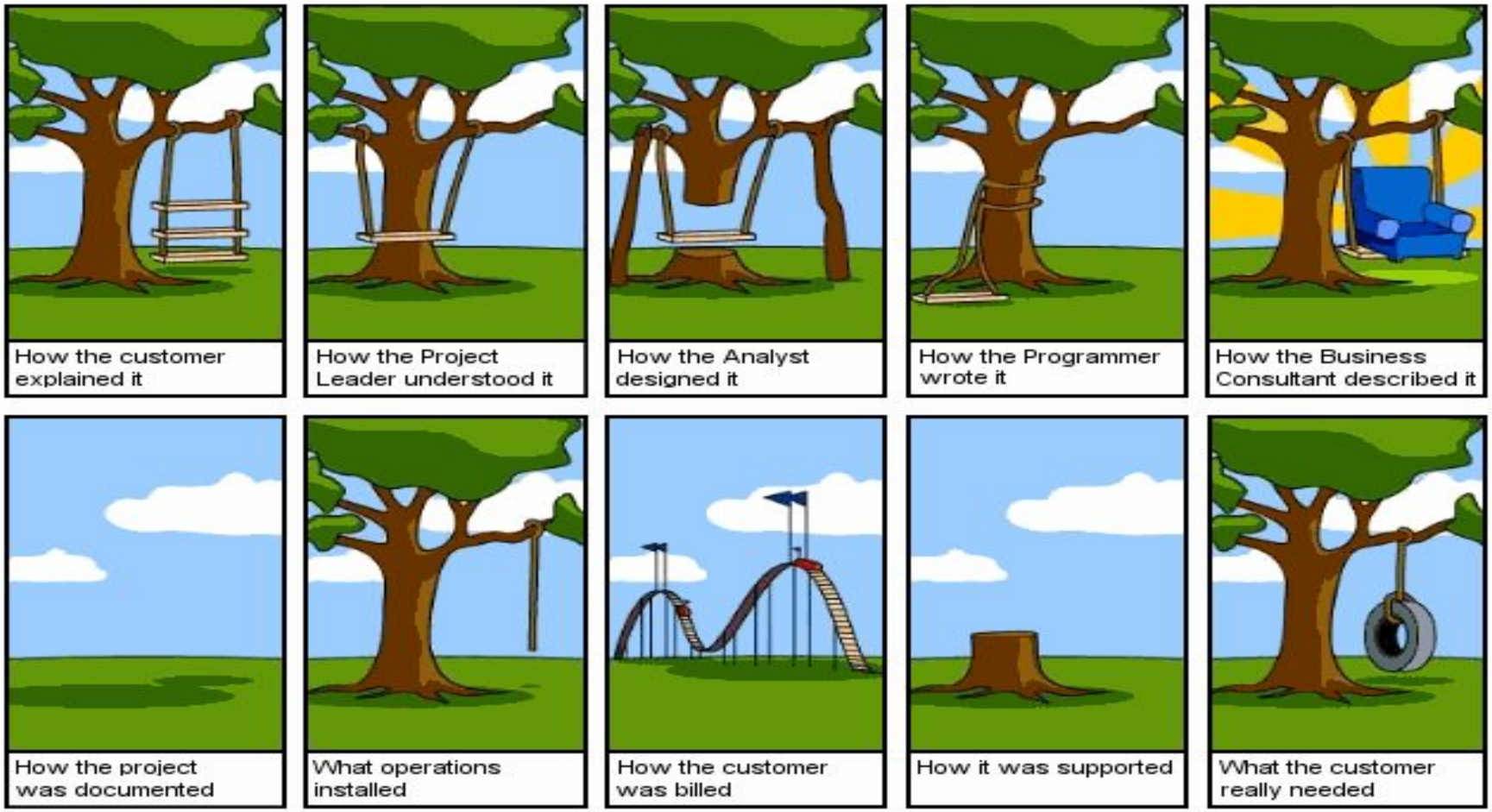
If possible,

Some guest lectures
from Practitioners
(excuse us later!)

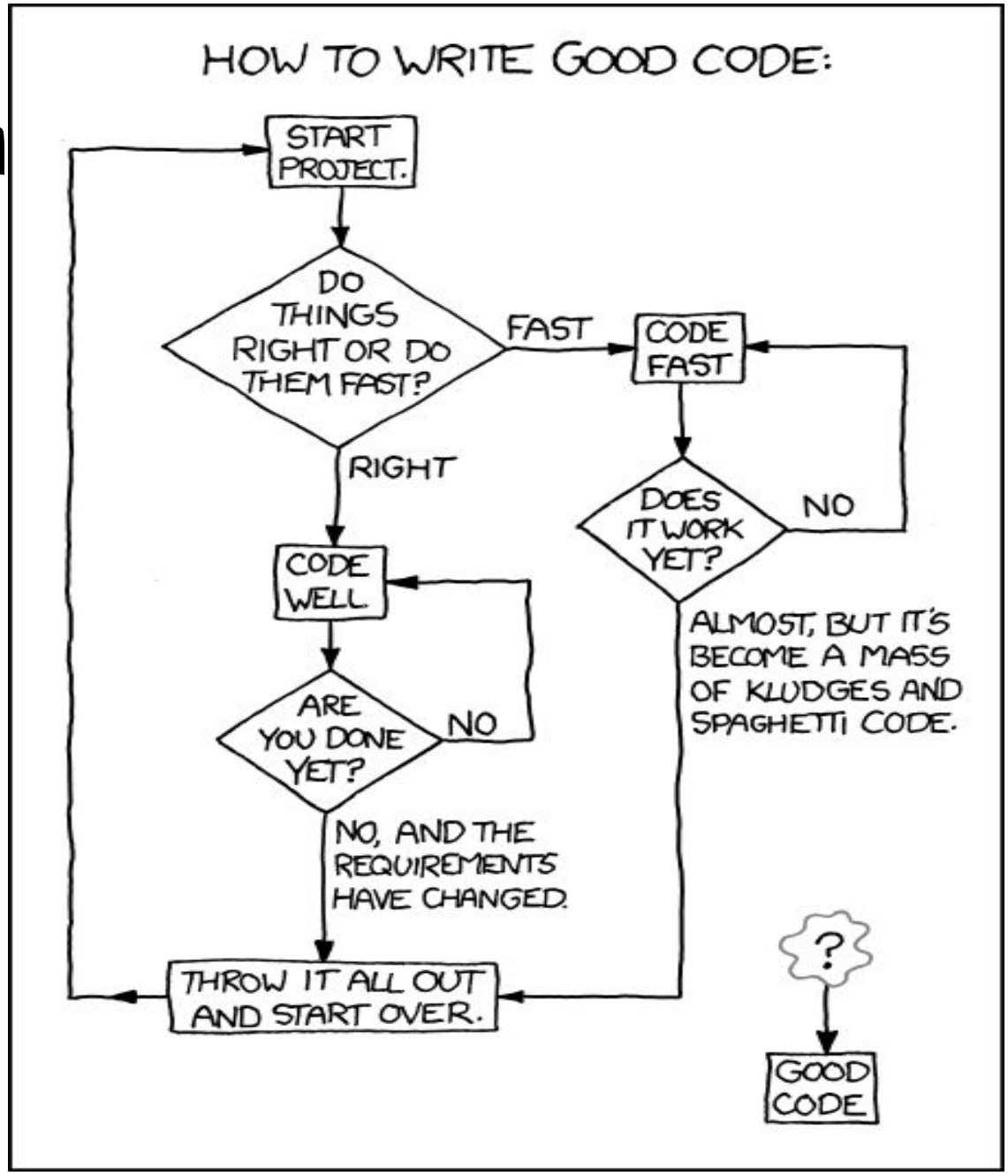
&

Some
articles, videos, terms, buzz words etc.

Not so positive,



More pessimism



And more... (or less?)

From the forward

Software developers are notorious for their impatience with anything that separates them from programming – *the medium has become the message.*

Symptomatically, the terms ‘hacking’ and ‘hacker’ have no equivalent in any other engineering discipline.

From the forward

Software developers are notorious for their impatience with anything that separates them from programming – *the medium has become the message.*

Symptomatically, the terms ‘hacking’ and ‘hacker’ have no equivalent in any other engineering discipline.

So what?

We are coders,
hackers,

We have our
laws!

Again from the forward, quoting

Hammurabi's code

229: If a builder build a house for some one, and does not construct it properly, and the house which he built fall in and kill its owner, then that builder shall be put to death.

Beginning with Chapter 1 of the textbook

The Nature of Software...

Software is intangible

- Hard to understand development effort

Software is easy to reproduce

- Cost is in its *development*

The industry is labor-intensive

- Hard to automate

Q: Is it like Avtar 1, 2 ?

The Nature of Software …

Untrained people can hack something together

- Quality problems are hard to notice

Software is easy to modify

- People make changes without fully understanding it

Software does not 'wear out'

Q: Why should we change it then?

The Nature of Software

Conclusions

- Much software has poor design and is getting worse
- Demand for software is high and rising

Q: Orkut *onwards*

- We are in a perpetual 'software crisis'
- We have to learn to 'engineer' software

We want you

Not to contribute to the crisis;

To be part of the solution, if possible.

Complexity

Economics

Psychology

As we end for the day, HW:

Watch a TedX Talk [Vineeth Paleri] and
Find out what happened to Infosys IT Project

Next class ?

Class 2, Class 3

Types of Software...

Custom

- For a specific customer [in-house or contracted]
- Meeting requirements is the main challenge

Generic

- Sold on open market
- Market research
- COTS (Commercial Off The Shelf)

Embedded

- Built into hardware
- Hard to change / replace

Types of Software

Differences among custom, generic and embedded software

	Custom	Generic	Embedded
Number of *copies* in use	low	medium	high
Total *processing power* devoted to running this type of software	low	high	medium
Worldwide annual *development effort*	high	medium	low

Types of Software - Another Classification

Real time software

- Control and monitoring systems
- Must react immediately
- Safety often a concern

Data processing software

- Used to run businesses
- Accuracy and security of data are key

Legacy software

What is Software Engineering?...

The process of solving customers' problems by the systematic development and evolution of large, high-quality software systems within cost, time and other constraints

What is Software Engineering?...

Solving customers' problems

This is THE GOAL
Buy or Build
Features [not bugs]
Effective Communication

What is Software Engineering?...

Systematic development and evolution

An engineering process involves applying *well understood techniques* in a organized and *disciplined* way

Many well-accepted practices have been formally standardized ISO/IEC/IEEE

Most development work is *evolution*

What is Software Engineering?...

Large, high quality software systems

software engineering techniques are needed because large systems *cannot be completely understood* by one person

Teamwork and co-ordination are required

Key challenge: Dividing up the work and ensuring that the parts of the system work properly together

The end-product must be of sufficient quality

What is Software Engineering?

Cost, time and other constraints

Finite resources

The benefit must outweigh the cost

Others are competing to do the job cheaper and faster

Inaccurate estimates of cost and time have caused many project failures

What is Software Engineering?...

The process of solving customers' problems by the systematic development and evolution of large, high-quality software systems within cost, time and other constraints

IEEE Definition: The application of a systematic, disciplined, quantifiable approach to the development, operation, maintenance of software; that is, the application of engineering to software.

Software Engineering and the Engineering Profession

The term Software Engineering was coined in 1968

People began to realize that the principles of engineering should be applied to software development

Engineering is a licensed profession

RMP
Licence to ?

Due to its intangible nature, software engineers do not require licence in order to code in most countries.

Software Engineering and the Engineering Profession

Ethics in Software Engineering

Software engineers shall

- Act consistently with public interest
- **Act in the best interests of their clients**
- Develop and maintain with the highest standards possible
- Maintain integrity and independence
- Promote an ethical approach in management
- Advance the integrity and reputation of the profession
- **Be fair and supportive to colleagues**
- **Participate in lifelong learning**

Stakeholders in Software Engineering

1. **Users**

   Those who use the software
   Objective: easy to learn & use,
   makes life easier and enjoyable.

2. **Customers**

   Those who pay for the software
   Objective: increase profits /
   run the business effectively.

3. **Software developers**

4. **Development Managers**

Software Quality Attributes

Usability

Users can learn it and fast and get their job done easily

Efficiency

It doesn’t waste resources such as CPU time and memory

Reliability

It does what it is required to do without failing

Maintainability

It can be easily changed

Reusability

Its parts can be used in other projects, so reprogramming is not needed

Software Quality and the Stakeholders

Customer:
solves problems at
an acceptable cost in
terms of money paid and
resources
used

User:
easy to learn;
efficient to use;
helps get work done

Developer:
easy to design;
easy to maintain;
easy to reuse its parts

Development manager:
sells more and
pleases customers
while costing less
to develop and maintain

Software Quality: Conflicts and Objectives

- The different qualities can conflict
  - Increasing efficiency can reduce maintainability or reusability
  - Increasing usability can reduce efficiency
- Setting objectives for quality is a key engineering activity
  - You then design to meet the objectives
  - Avoids ‘over-engineering’ which wastes money
- Optimizing is also sometimes necessary
  - Reliability is very expensive
  - Warranty T&C

Internal Quality Criteria

These:

- Characterize *aspects of the design* of the software
- Have an effect on the external quality attributes

Examples:

- The effort or software architecture
- The amount of commenting of the code
- The use of efficient algorithms

Short Term Vs. Long Term Quality

Short term:

- Does the software *meet the customer's immediate needs*?
- Is it sufficiently efficient for the volume of data we have *today*?

Long term:

- Maintainability
- Customer's future needs
- Can the software handle larger volumes of data?
  - Does it scale?