Introductory Session Continued

January 2023

What kind of project you expect?

What kind of project you expect?

All New

Almost New

Quite Old

Types of SWE Projects

Software Engineering Projects - Maintenance

Most projects are *evolutionary* or *maintenance* projects, involving work on *legacy* systems

> Corrective projects: fixing defects
>
> Adaptive projects: changing the system in response to changes in
>
> > Operating system
> >
> > Database
> >
> > Rules and regulations
>
> Enhancement projects: adding new features for users
>
> Reengineering projects: changing the system internally so it is more maintainable

Software Engineering Projects - NEW

'Green field' projects

New development
The minority of projects
Not easy to be part of one in the Indian context

Except

Joining / Starting a start-up :-)

And of course,

Your BTP

Motivation!

TBI, NITC

KSUM

Motivation!

It is better to
TRY and FAIL
than
FAIL to TRY

Software Engineering Projects – The Trend

Projects that involve building on a *framework* or a set of existing components.

A framework is an application that is missing some important details.

React , Angular JS
Django, **Flask**, Ruby on Rails

Tools like GWT

It allows developers to avoid needless coding, accelerate program building with stable working set of standard features, like data storage, mailing or printing, and economise your time and money, especially when you need to make your app available on different platforms.

Software Engineering Projects – The Trend

Projects that involve building on a *framework* or a set of existing components.

Such projects:

==Involve plugging together *components* that are:==
==Already developed.==
==Has large functionality.==

Benefit from reusing reliable software.

Provide much of the same freedom to innovate found in green field development.

OpenAI Codex

Activities Common to SWE Projects
or
SDLC

Custom VS Green Field Project

Project Planning
Requirements Gathering, Analysis, **Specification**
Design and Modelling
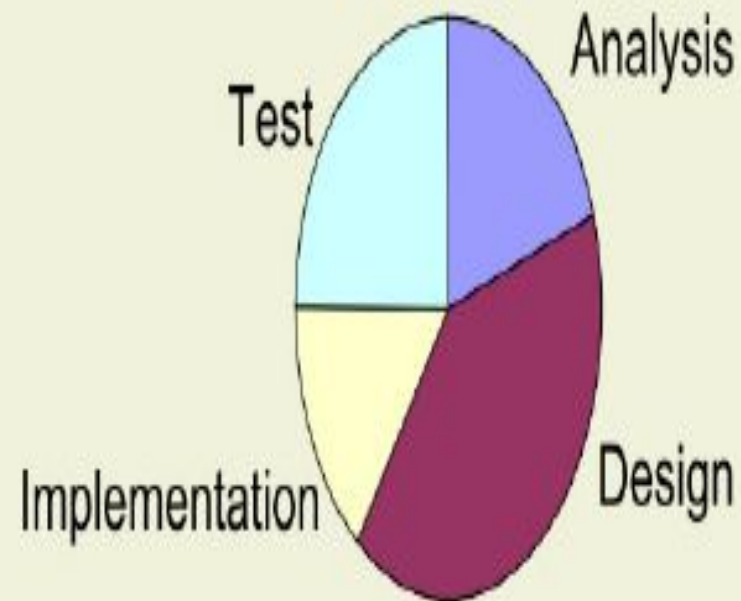**Implementation** [Coding/Programming]
Testing
Deployment
Maintenance

Quality Assurance

Effort Distribution – Two Different Takes

- Analysis   20%
- Design   40%
- Implementation   15%
- Test   25%

Project Planning

First Ask: Do we need to take this up?

Feasibility Analysis

Cost Estimations

Organizational Planning
  People
  Schedules & Deadlines

Requirements Specification

Domain analysis

- Need for domain expertise

Defining the problem

Requirements gathering

Obtaining input from as many sources as possible

Requirements analysis

Requirements specification

- <mark>Writing detailed instructions about how the software should behave</mark>
- Precise & Non-Conflicting
- No implementation Details - Just WHAT not HOW

SRS

Software Requirements Specification
Clear
Concise
Concrete

A software requirements specification (SRS) is a document that describes what the software will do and how it will be expected to perform.
  [*not HOW the software is developed*]

It also describes the functionality the product needs to fulfill all stakeholders (business, users) needs.

SRS – like construction contract

Avoid conflicts, always have a written Construction Contract

Activities Common to Software Projects DESIGN

Deciding how the requirements should be implemented, using the available technology

Systems engineering: Deciding what should be in hardware and what in software

Software architecture: Dividing the system into subsystems and deciding how the subsystems will interact

Detailed design of the internals of a subsystem

User interface design

Design of databases

Activities Common to Software Projects

Implementation

Testing

Deployment

Maintenance Maintenance Maintenance Maintenance
Maintenance Maintenance Maintenance Maintenance

Activities Common to Software Projects

Quality Assurance [not a separate phase]

Reviews and inspections

Testing

TESTING [Automation Scripts]

MAINTENANCE [Well Designed, Easily Maintainable]