

# Introduction To Software Engineering

## Outline:

1. Nature of Software Project
2. Engineering Approach
3. Software Process
4. A process Step
5. Characteristics of a good process.

# Introduction To Software Engineering

## Challenges in large project

1. Effort Intensive
2. High Cost
3. Large Development Time
4. Changing needs for user
5. High risk of failure , user acceptance , performance , maintainability

# Why Software Engineering?

- Late 1960 H/W prices were falling but S/W prices rising
- Many Software Project failed
- They did not satisfy the requirements.



■ ■ ■ ■ ■

- Larger SP require larger Team
- Complexity of software project increased.
- Larger software system are more difficult & expensive to maintain
- Demand of new S/W increased faster than ability to generate new Software
- All of the above are the attributes of “**Software Crisis**”

The term software “Software Engineering” First introduced at a conference in late 1960 to discuss the “Software Crisis”

# Successful Software System

- Software development project have not always been successful
- When do we consider a software application successful?
  - Development Completed
  - It is use full
  - It is Usable
  - It is Used



## Some inspiration quote:

- “Ideas are the building blocks of ideas”-Jason
- “Computer make it easy to do a lot of things, but most of the things they make it easier to do don’t need to be done”- Andy Rooney
- “There is no computer that has common sense”-Marvin Minsky
- “You can’t always predict ,but you can always prepare”-

# Reasons for failure

- Schedule slippage
- Cost overruns
- Does not solve user's problem
- Poor quality of Software
- Ad-hoc software development results in such problems
- No planning of development work

# Reasons.....

- Deliverables to user not identified
- Poor understanding of user requirement
- No control or review
- Technical incompetence of developers
- Poor understanding of cost & effort by both developers & user.



# Engineering :Other Disciplines

- Large projects common & successfully done
  - Building bridges,dams
  - Power Plants
  - Aircraft,missiles



# “Engineering” a solution-

- To design,develop(build,fabricate) an artifact that meet specifications efficiently ,cost effectively & ensure quality
- Using scientific principles
- Cycles: Specification Cost Effective

Good Quality



# “Engineering” a solution.....

- Require well-define approach,repeatable,predictable
- Large projects requires managing the project itself
- Manage people ,money,equipment,schedule
- Scale makes big Difference: a hut,2 storeyed building  
,50 storeyed building

# “Engineering” a solution-

- Quality extremely important : relates to failure,efficiency,usabilty
  - People willing to pay for quality
    - E.g: buying T.V

# Large project:

- Involve different typrs of people:
  - Architect,civil engineer,electrical engineer.
- Continuous Supervision for quality assurance:
  - On site supervision(check cement,steel quality...)
- Many delivarables: architecture plan,model,structure diagrams,electrical cabliing,layout..



# Large project:

- Standards, regulation. conventions need to be followed.
- Steps ,milestones defined & reviews are carried out: progress is visible

# Software Project

- Software is different from the other product:
  - Cost of production concentrated in development
  - Maintenance consists of making corrections && enhancing or adding functions
  - Progress in development is difficult to measure:  
80% or 90% complete.

# Apply Engineering approach:

- Planning & control even more important in software development.
  - Attempt to estimate cost/effort
  - Plan & schedule work
  - Involve user in defining requirements
  - Identify stages in development
  - Define clear milestones so that progress can be measured



# Apply Engineering approach:

- Schedule reviews both for control & quality
- Define deliverables
- Plan extensive testing

# What is Software:

- Software is programs to provide functions and performance of desire tasks.
- Collection of computer programs, procedures, rules & associative documentation & data
- Computer instruction written to be executed on hardware, including OS, utility programs & application programs

# What is Engineering

- Engineers try to discover problems and find / develop and apply appropriate theory, methods, tools, to the solution of the problems

# What is Software Engineering?

- “SE is concerned with software system developed by teams rather than individual programmers, uses engineering principles in the development of these system and is made up of both technical & non-technical aspects.” -: (Sommerville)
- “SE is a discipline that integrates method ,tools & procedure for the development of computer software”-: (Pressman)
- “The systematic approach to the development , operation , maintenance,& retirement of software” :- (IEEE)

# What issues are considered in Software Engineering?

- What is the problem to be solved
- What are the features of the entity that are used to solve the problem
- How will the solution be realized
- What approaches will be used to uncover errors in the design and construction of the entity
- How entity will be maintained

# Software Application

## ● System Software

- Programming Languages: Basic, C, C++
- Operating System: WINDOWS , LINUX, UBUNTU
- Utilities: They expand the performance of the OS by adding functions that are not part of the original OS. File conversion, data compression , defragmentation are the examples of utility software.

## ● Application Software

- Spreadsheets
- Database Management
- Word Processing
- Games & Entertainment
- Internet

# Software Errors

- What are the reasons:
  - Specification may be wrong
  - The system design may be wrong
  - The program design design may be at fault
  - The program code contains error

# Types of errors:-

- Algorithmic error
- Syntax error
- Computation & precision error
- Documentation error
- Stress of overloaded error
- Capacity or boundary error

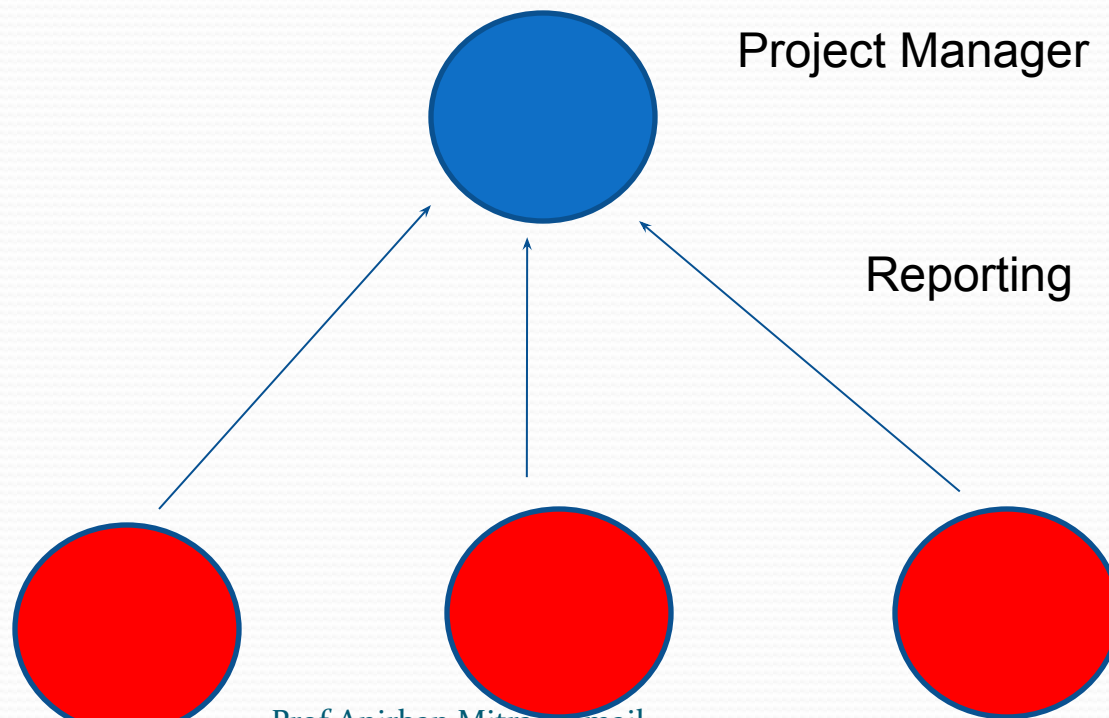


# Job of Software Developer

- Dealing with user :
  - Concern with ease of use & response time.
- Dealing with technical people:
  - Concerned With coding , databases , file structure etc.
- Dealing with Management:
  - Concerned with return on their investment.
  - Cost-Benefit analysis.
  - Schedule

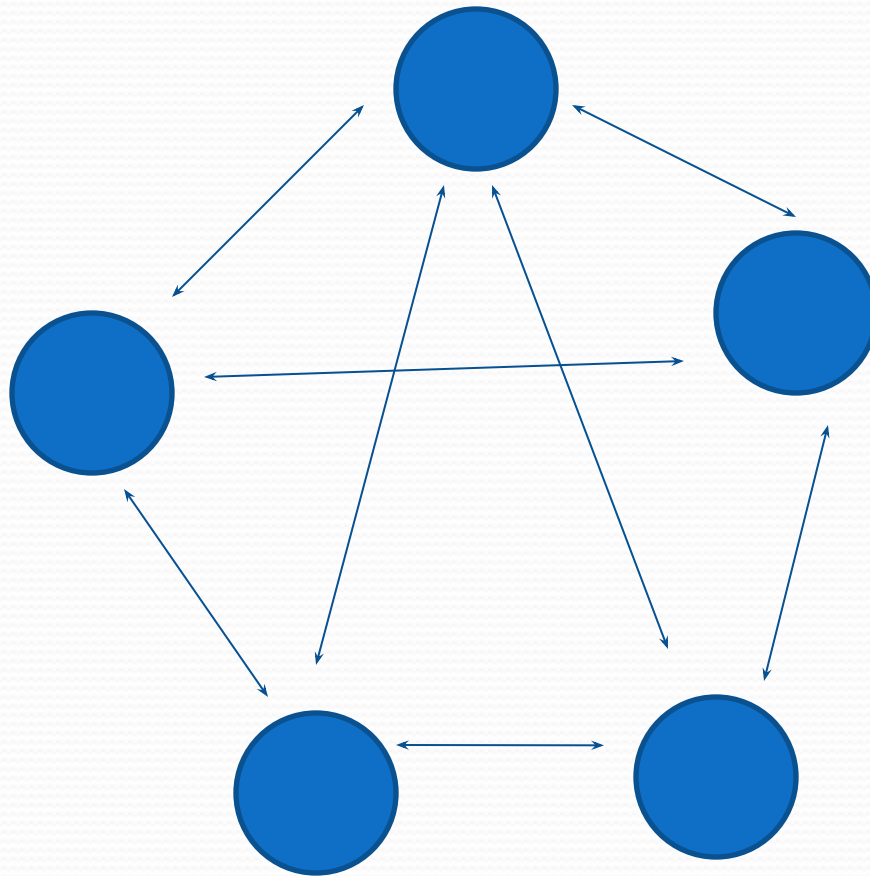
# Organization & Team Structure

## Chief Programmer Team

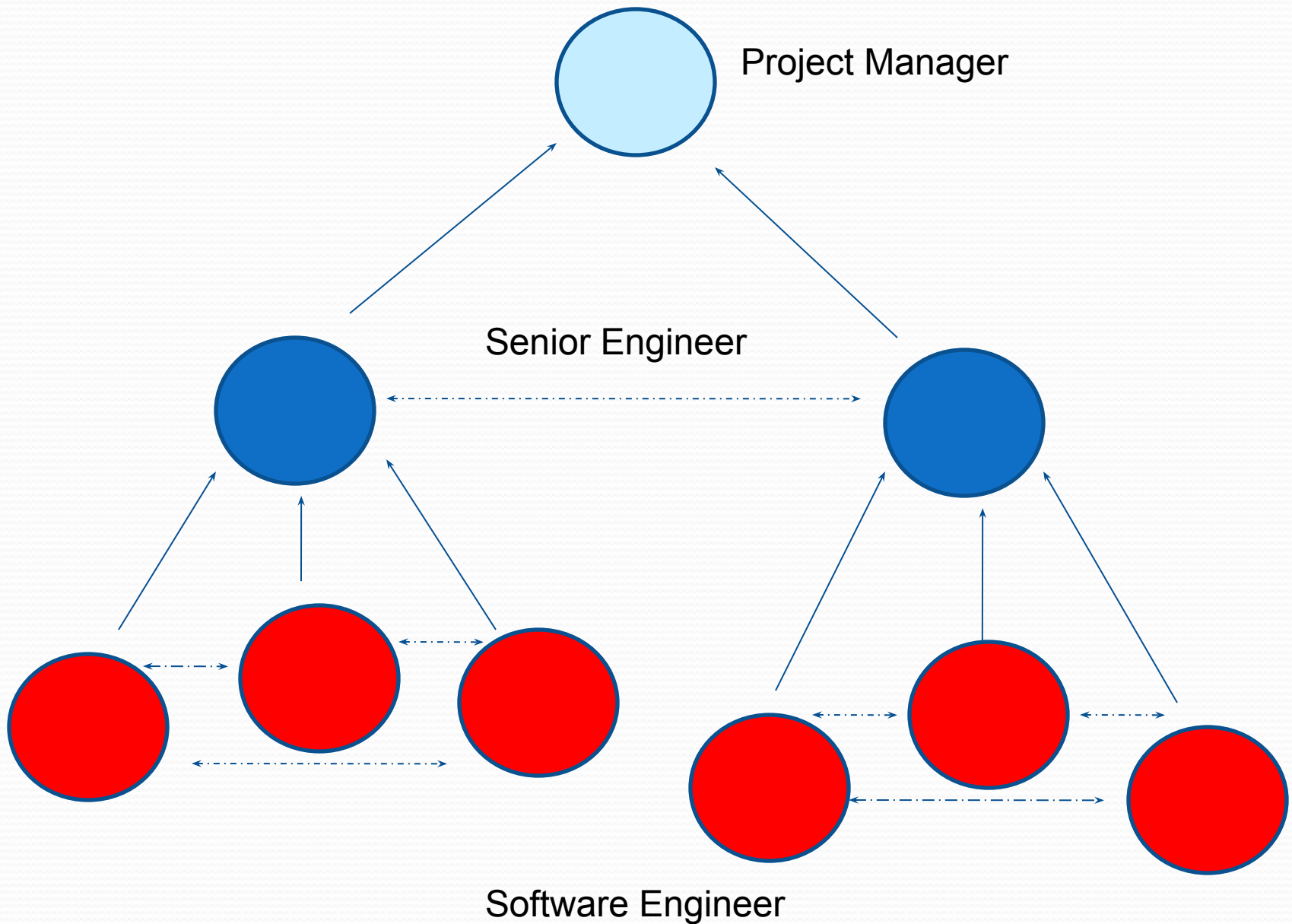


Prof Anirban Mitra : e-mail:  
anirban.mitra@pmail.com  
Contact:9830312660  
**Software Engineers**

# Democratic Team



# Mixed control team Organization

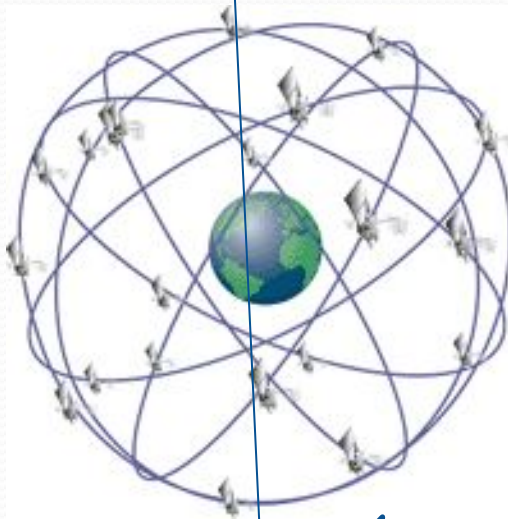


# Risk Management

- A risk is any anticipated unfavorable event or circumstance that can occur while a project is underway
  - Risk identification
  - Risk Assessment
  - Risk Containment.

# Some possible risk are:

- Experienced staff leaving the company before completion
- Change in Technology
- Change in requirement
- Change in Government Policy
- Completion from other company
- Financial Problem
- Underestimating cost & effort



# Risk assessment

- Rank the risk in terms of their damage causing potential

- $P = r * s$

- $P$  : priority with which the risk must be handled
- $r$  : probability of the risk becoming true
- $s$  : severity of the damage due to that risk



# Risk Containment

- Avoid the risk :
  - Discussion with customer to change requirement/  
Incentives
- Transfer The Risk:
  - Insurance Cover
- Risk Reduction:
  - If there is risk that some key personal might leave, new recruitment may be planned

# Risk leverage

- To choose between the different strategies of handling a risk, The Project Manager have to compute the **risk leverage**.

RL =

$$\frac{(\text{Risk exposure before reduction} - \text{Risk exposure after reduction})}{\text{Cost Of Reduction}}$$