SOFTWARE PROJECT MANAGEMENT

LECTURE # 2

4 P's in Project Management

Engr. Ali Javed

Contact Information

- Instructor: Engr. Ali Javed

  Lecturer

  Department of Software Engineering

  U.E.T Taxila

  - Email: ali.javed@uettaxila.edu.pk
  - Contact No: +92-51-9047592
  - Office hours:
    - **Monday, 11:00 - 01:00, Office # 7**

Course Information

- **Course Name: Software Project Management**
- **Course Code: SE-401**
- **CMS Link:** http://web.uettaxila.edu.pk/CMS/AUT2011/seSPMbs/index.asp

Recommended Websites

www.maxwideman.com
www.4pm.com
www.bettermanagement.com
www.michaelgreer.com
www.e-programme.com
www.managementhelp.org
http://project-management.informationeering.com
www.comp.glam.ac.uk
http://www.softwareprojects.org
www.columbia.edu
www.projectmanagement.ittoolbox.com

Recommended Websites

www.4pm.com
www.ics.ltsn.ac.uk
www.pmboulevard.com
www.mapnp.org
www.projectconnections.com
www.niwotridge.com
www.spmn.com
www.processimpact.com
www.projectperfect.com
www.pmtoday.co.uk
www.projectability.co.uk
www.easyprojects.net

www.methods-tools.com
www.pmforum.com
www.focusedperformance.com
www.tenstep.com
www.managementhelp.com

4 P's in PM Spectrum

- People
- Product
- Process
- Project

People [1]

- Stakeholders
- Team Leaders
- Software Team
- Agile Teams

People

- The most important factor in success of software project.
- "Companies That sensibly manage their investment in people will prosper in the long run" Tim & Tom.
- Cultivation of motivated and highly skilled software people has always been important for software organizations.
- The "people-factor" is so important that SEI has developed People Management Capability Maturity Model (PM-CMM).

PM-CMM

- Developed by SEI
  - ✓ "To enhance the readiness of s/w organizations to undertake increasingly complex applications by helping to attract, grow, motivate, deploy, and retain the talent needed to improve their software development capability"
  - ✓ In simple words - to enhance the people's capabilities through personnel development
- Organizations that achieve high levels of maturity in PM-CMM have a higher likelihood of implementing effective software engineering practices

PM-CMM

- Key Practice Areas of PM-CMM
  - ✓ Recruiting [3]

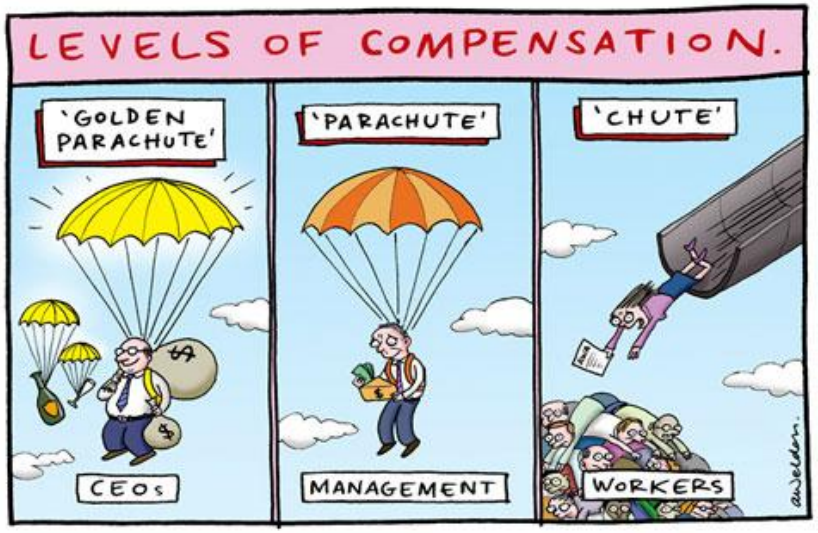
  - ✓ Selection [3]

  - ✓ Performance Management [4]

  - ✓ Training

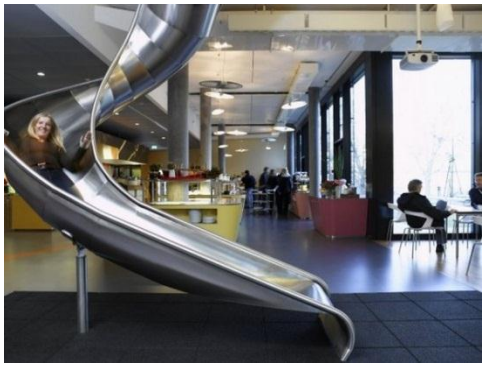
PM-CMM

- Key Practice Areas of PM-CMM
  - ✓ Compensation [5]
  - ✓ Organizational design [6]
  - ✓ Career development [7]
  - ✓ Team/culture development [8]



PM-CMM

- Key Practice Areas of PM-CMM
  - ✓ Work environment



Peoples involved in Software Process

- Stakeholders
- Team Leaders
- Software Team
- Agile Teams

The Stakeholders

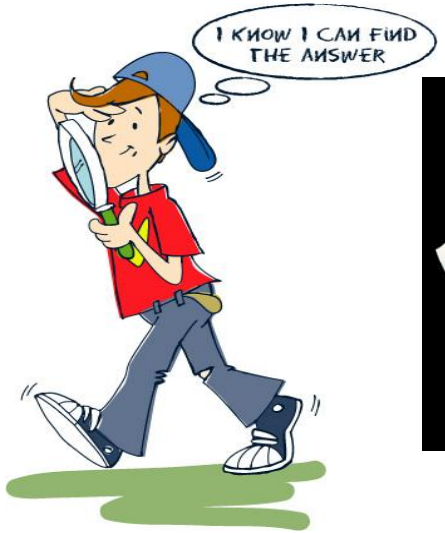
- They can be categorized into one of the following
  - ✓ Senior Managers
    - they define business issues that often have significant influence on business
  - ✓ Project (technical) managers
    - they must plan, motivate, organize and control the practitioners who do software work
  - ✓ Practitioners
    - They deliver the technical skills necessary to engineer a product or application
  - ✓ Customers
    - They specify the requirements for the software to be engineered
  - ✓ End Users
    - They interact with the software after it is released for production use

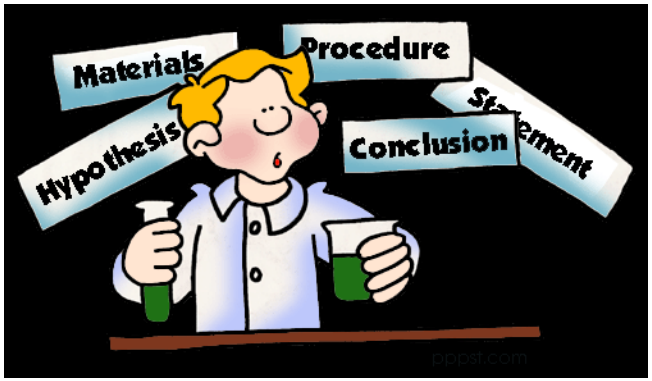
The Team Leaders

- Competent Practitioners often make poor team leaders as they lack the right mix of skills
- In his excellent book of technical leadership, **Jerry Weinberg** suggests a **MOI model** of leadership MOI Model of Leadership
  - ✓ Motivation
    - ❖ encourage technical people (by "push" or "pull") to produce
  - ✓ Organization
    - ❖ Apply , improve processes efficiently
  - ✓ Ideas or Innovation
    - ❖ Make people feel creative
    - ❖ Be Creative

The Team Leaders

- **The Team Leaders** - Characteristics of an effective project managers:





  - ✓ Problem Solving
    - ❖ Diagnostic
    - ❖ Skill to solve
    - ❖ Ability to design solution

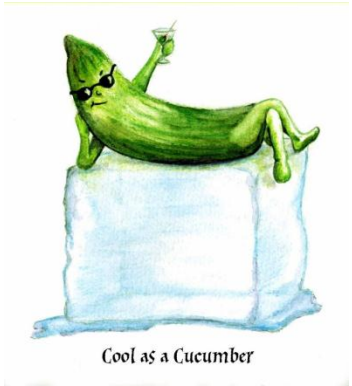
  - ✓ Managerial Identity
    - ❖ Control the project

The Team Leaders

- **The Team Leaders** - Characteristics of an effective project managers:

  - ✓ Achievement
    - ❖ Reward Initiative
    - ❖ Encourage Controlled risk taking

  - ✓ Influence and team building
    - ❖ Influence the team
    - ❖ Read people's mind and respond according to their needs
    - ❖ Be controlled in stress situations

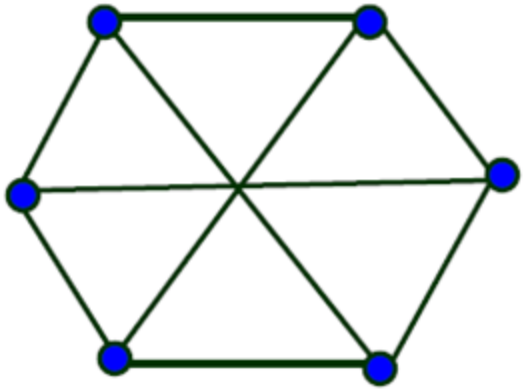


The Software Teams

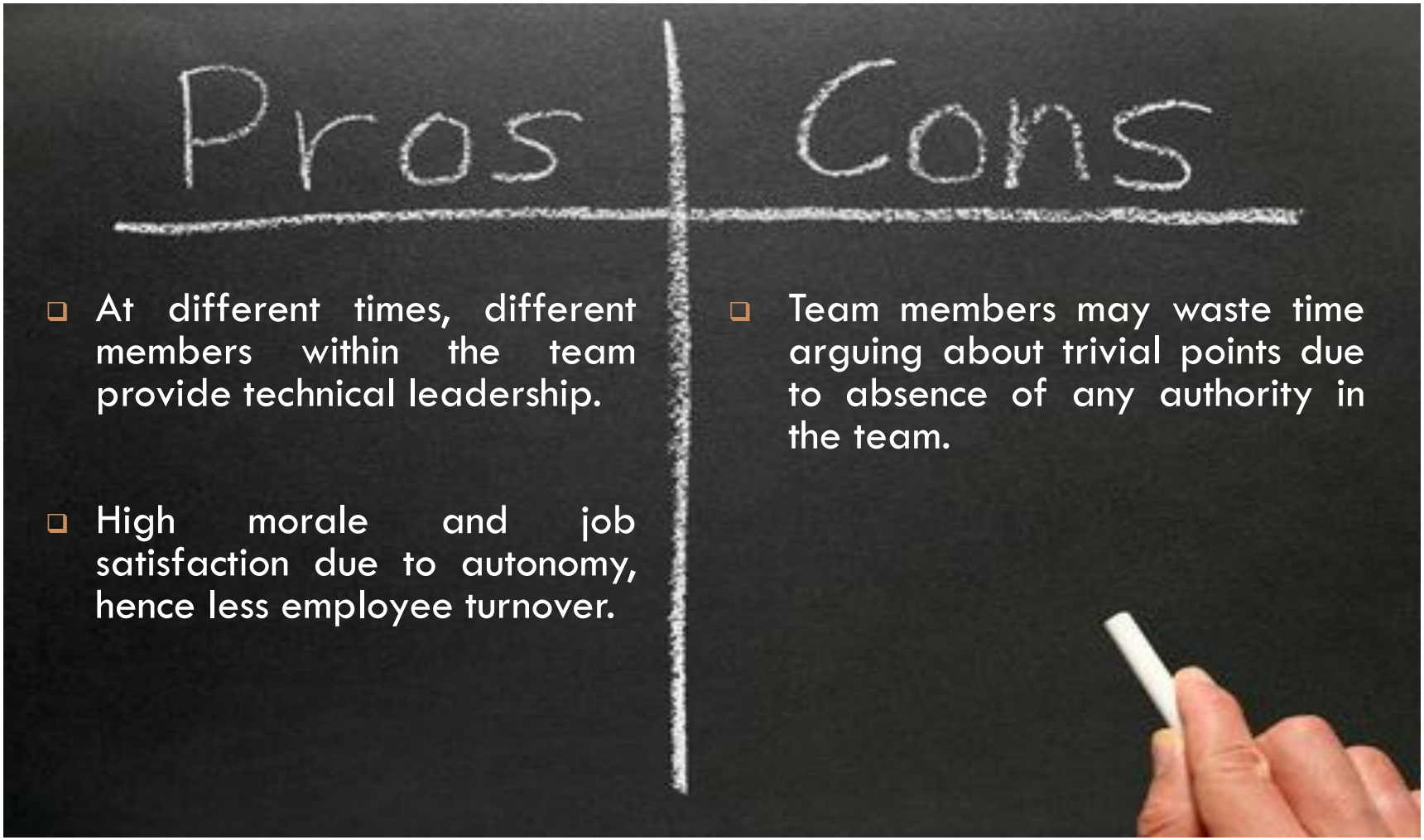
- Organizations/Structure of teams:
  - ✓ Democratic decentralized
  - ✓ Controlled decentralized
  - ✓ Controlled centralized

The Software Teams: Democratic decentralized

- Democratic decentralized
  - ✓ No permanent leader
  - ✓ Communication is horizontal
  - ✓ Suitable for small projects requiring less than 5 to 6 engineers, research-oriented projects

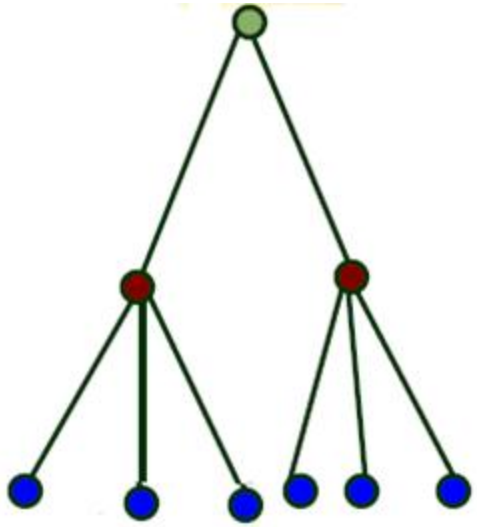


The Software Teams: Democratic decentralized

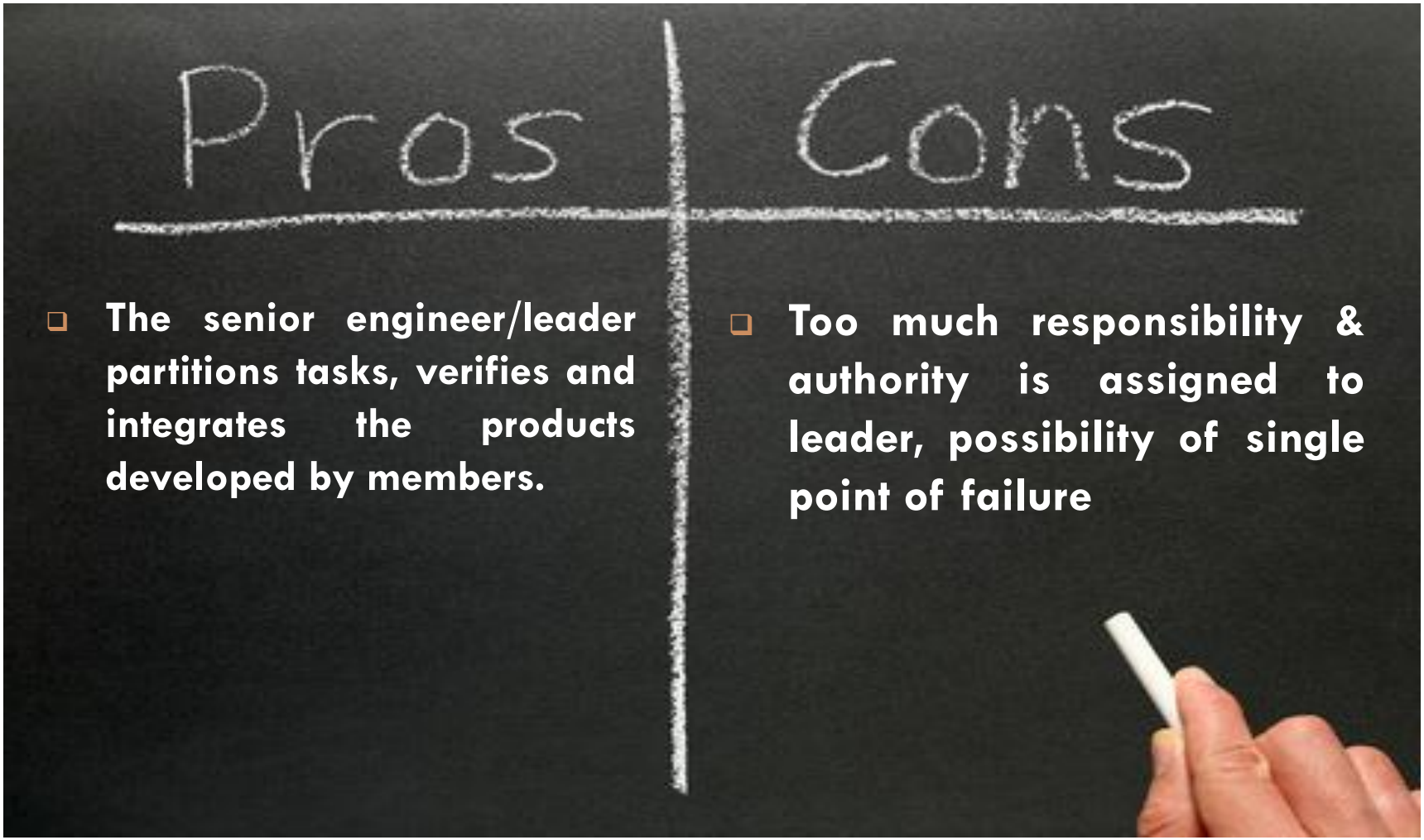


The Software Teams: Controlled Centralized

- Controlled centralized
  - ✓ Defined team leader
  - ✓ Problem solving , communication and management by team leader
  - ✓ Communication is vertical

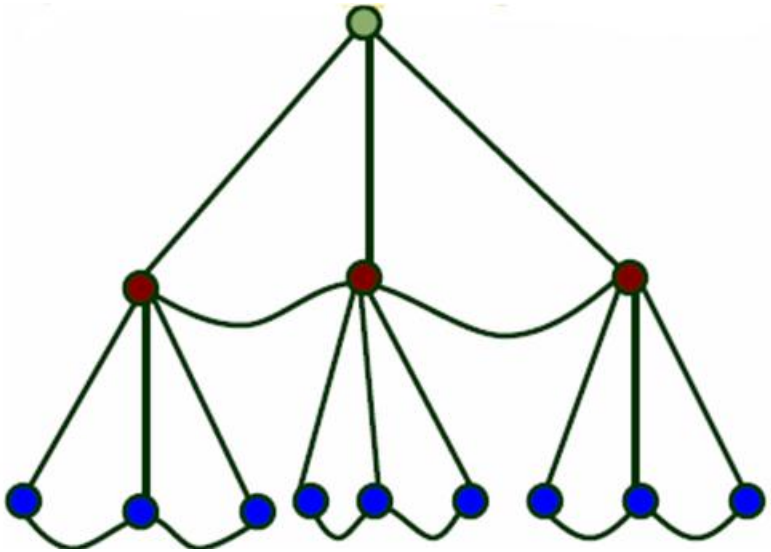


The Software Teams: Controlled Centralized



The Software Teams: Controlled Decentralized

- Controlled decentralized
  - ✓ Draws upon the ideas from both earlier structures
  - ✓ Defined Leader
  - ✓ Horizontal communication
  - ✓ Problem solving is a group activity
  - ✓ Suitable for large organizations

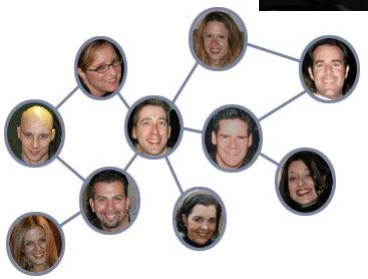


The Software Teams

- *Mantei* describes seven factors that should be considered when planning team structure:
  - ✓ Difficulty of task
  - ✓ Size of resultant code (no. of lines)
  - ✓ Time that team will stay together
  - ✓ Degree of modularization
  - ✓ Required quality and reliability of the system being built
  - ✓ Rigidity of delivery date (schedule)
  - ✓ Degree of communication

Communication & Coordination Issues

- Formal approaches
  - ✓ Writings (SE documentation, Customer requests, etc.)
  - ✓ Status review meetings
  - ✓ Design and code inspections

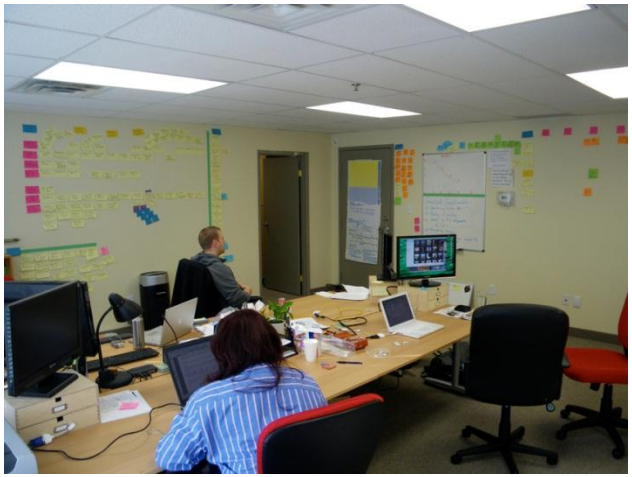


- Informal approaches (more personal)
  - ✓ Interpersonal networking [9]
  - ✓ Sharing of ideas on ad hoc basis
  - ✓ Seeking help from inside or outside the project team when problem arises

- Electronic Communication
  - ✓ E-mail, electronic bulletin boards [10], video conferencing

The People - Agile Teams

- *Agile software development encourages customer satisfaction and early incremental delivery of software with overall simplicity.*
- **Agile teams** are small, highly motivated teams.
- They adopt many characteristics of successful software project teams and avoid toxins that create problems.
- They are self organizing and do not necessarily maintain a single team structure
- Agile process models give significant autonomy to agile teams.



The People - Agile Teams

- Planning is kept to minimum.
- The agile team is allowed to select its own approach (e.g., process, methods, tools).
- The agile team may have daily team meetings to coordinate and synchronize the day's work.
- With each passing day, this self organization and collaboration move the team towards a completed software increment.

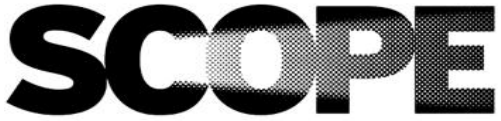
The Product [1]

- Software Scope
- Problem Decomposition

The Product

- The product and the problem it is intended to solve must be examined at very beginning of the software project.
- The **scope** of product must be established and bounded.
  - ✓ Bounded scope means
    - establishing quantitative data like no. of simultaneous users, max. allowable response time. etc.
    - Constraints and limitations
    - and mitigating factors described
- The **problem** that the product is addressing must be decomposed

Software Scope



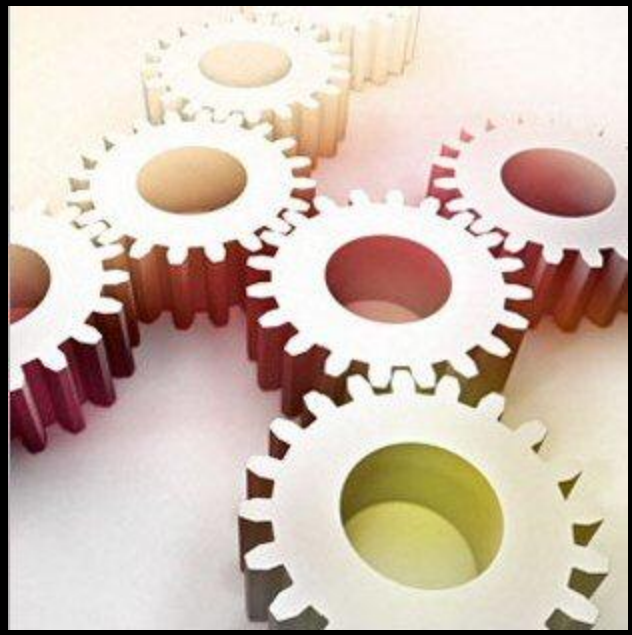
- **Scope is defined by**
  - ✓ **Context**
    - Functional location of the software product into a large system, product or business context
    - Constraints involved
  - ✓ **Information Objectives**
    - What data objects are required as i/p or o/p
  - ✓ **Function and Performance**
    - What function does the software system perform on i/p to produce o/p
    - What level of performance is required

Problem Decomposition

- Also called partitioning OR problem elaboration
- This activity is at core of requirements analysis
- Divide and conquer policy for complex problems
- A complex problem is partitioned into smaller problems that are more manageable.
- Decomposition make planning easier.
- Decomposition in 2 major areas
  - ✓ Functionality that must be delivered
  - ✓ Process that will be used to deliver product

The Process [1]

- Process
- Framework Activities
- Process Models
- Process Decomposition



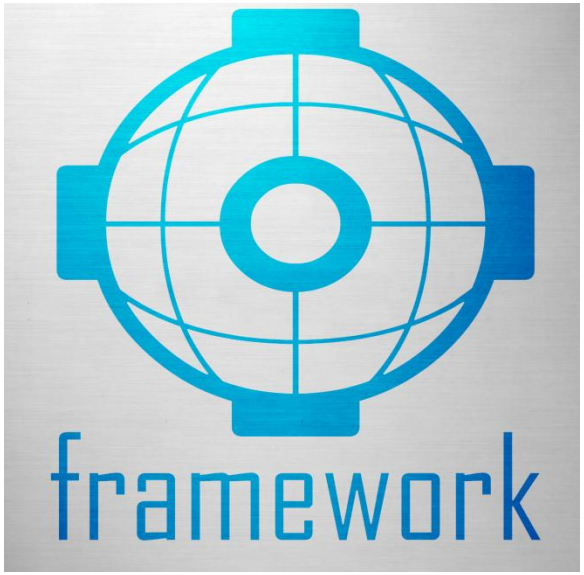
The Process

- A software process provides the framework from which a comprehensive plan for software development can be established.
- Common process framework activities which are applicable to all software projects are:

  - ✓ Communication
  - ✓ Planning
  - ✓ Modeling
  - ✓ Construction
  - ✓ Deployment

Common Process Framework Activities

- These characterize a software process and are applicable to all software projects
  - ✓ Communication
  - ✓ Planning
  - ✓ Modeling
  - ✓ Construction
  - ✓ Deployment



- These are applied to software engineering work tasks (e.g., different product functions)

The Process Models

- Different process models:
  - ✓ Linear sequential, Prototyping, RAD, Spiral, Formal …
- Project manager must decide about which model to use depending on
  - ✓ Customers who have requested the product
  - ✓ People who would work on project
  - ✓ Product characteristics
  - ✓ Project environment
- Project planning begins once model is selected

Process Decomposition

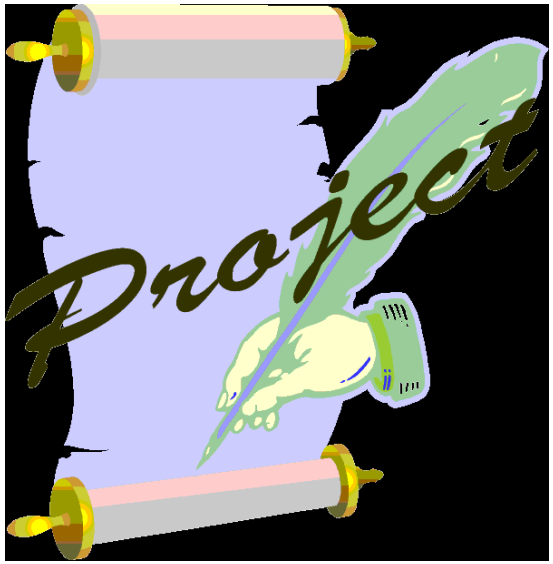
- The way a process is decomposed depends on project complexity
- Decomposition involves outlining of work tasks involved in each process framework activity
- Example of decomposition for 'communication' activity for a simple project:
  - ✓ Develop a list of clarification issues
  - ✓ Meet with customer to discuss clarification issues
  - ✓ Jointly develop statement of scope
  - ✓ Review the statement of scope with all concerned
  - ✓ Modify the statement of scope id required

The Project [1]

- Project
- Signs of Projects Risk
- How to Avoid Project Risks

The Projects

- The software projects must be planned and controlled effectively to avoid complexities.
- The project managers and engineers must understand the critical success factors and develop a common sense approach for planning, monitoring and controlling the project.



Signs of Projects Risk

- *John Reel* describes ten signs that indicate that project is in jeopardy:

  - ✓ Software people don't understand customer needs
  - ✓ Product scope is poorly defined
  - ✓ Changes are managed poorly
  - ✓ The chosen technology changes
  - ✓ Business needs change
  - ✓ Deadlines are unrealistic
  - ✓ Users are resistant
  - ✓ Sponsorship is lost
  - ✓ Team lacks skills
  - ✓ Managers avoid best practices

How to avoid problems?

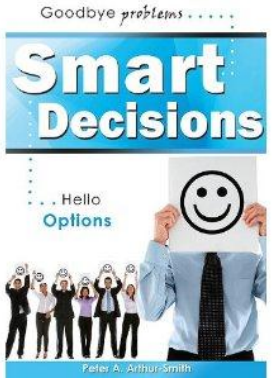
- Start on the right foot
  - ✓ Involves detailed understanding of project
  - ✓ setting realistic objectives & expectations
  - ✓ Selecting the right team
  - ✓ Facilitating the team

- Maintain Momentum
  - ✓ Provide incentives
  - ✓ Reduce bureaucracy and give autonomy to team members but with supervision

- Track Progress
  - ✓ Assess progress as work products are produced

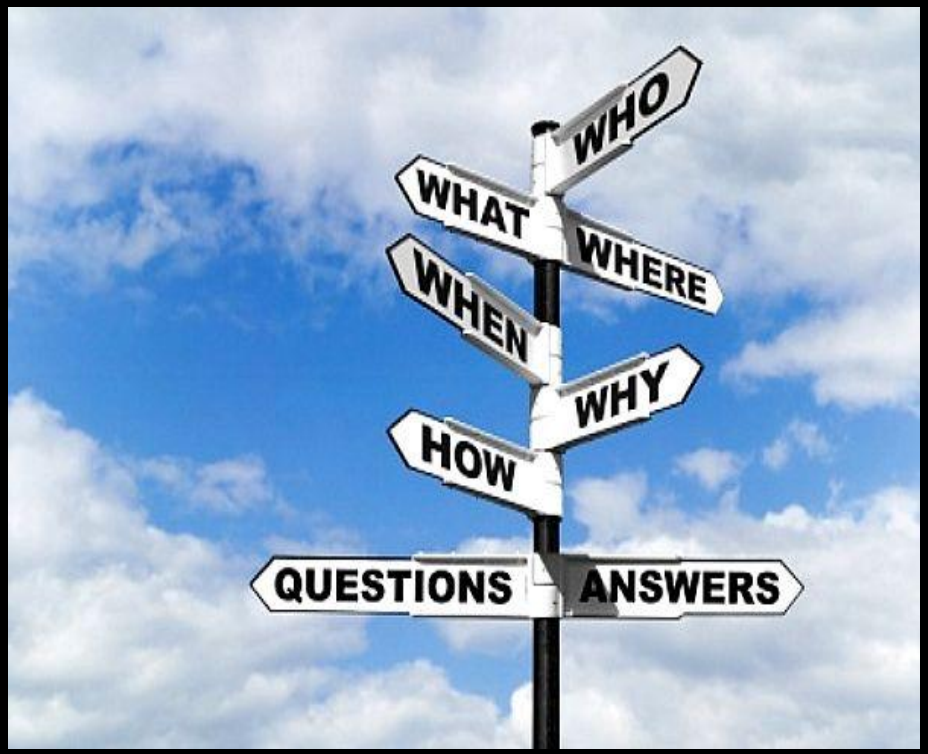
How to avoid problems?

- Make smart decisions
  - ✓ When possible, use existing software components / COTS software
  - ✓ Choose standard approaches and keep it simple
  - ✓ Avoid risks and allocate more time than needed for complex/risky tasks



- Conduct a postmortem analysis
  - ✓ Compare planned and actual schedule
  - ✓ Collect and analyze project metrics (standards)
  - ✓ Get feedback from team and customers
  - ✓ Establish record of lessons learnt for each project

W5HH Principle



W^{5}HH principle

- Suggested by Barry Boehm in one of his papers
- Excellent **planning outline** for project managers and software team
- Applicable to all sizes of software projects
- It is an approach to address
  - ✓ project objectives
  - ✓ Milestones & schedule
  - ✓ Responsibilities
  - ✓ Management & technical approaches
  - ✓ Required resources

W^5HH principle

- **W**hy is the system being develop?
  - ✓ Answer to this questions help assess validity of business reason for the software work.
  - ✓ It answers if the business purpose justifies the expenditure of people, time and money
- **W**hat will be done?
  - ✓ Answer to this question establishes the task set required for project
- **W**hen will it be done?
  - ✓ Answer to this question helps the team establish a project schedule by identifying when tasks have to be conducted and when milestones are to be reached

W^5HH principle

- **W**ho is responsible for a function ?
  - ✓ Answer to this question establishes roles and responsibility of each team member
- **W**here are they organizationally located ?
  - ✓ Answer to this question indicates that all roles and responsibilities are not limited to the software team itself, the customers, users and stakeholders also have responsibilities.
- **H**ow will be job done technically and managerially ?
  - ✓ Once product scope is establishes, a technical and management strategy must be defined for it.
- **H**ow much of each resource is needed ?
  - ✓ Answer to this question is derived by developing estimates based on answers to earlier questions.

References

1. Software Engineering by Roger Pressman
2. Team Structures Portion taken from Gary Pollice lectures on Software Project Management
3. http://recruitment.naukrihub.com/recruitment-vs-selection.html
4. http://en.wikipedia.org/wiki/Performance_management
5. http://en.wikipedia.org/wiki/Workers%27_compensation
6. http://www.inovus.com/organiza.htm
7. http://www.wisegeek.com/what-is-career-development.htm
8. http://pathikconsulting.com/team-and-culture-development-corporate-culture-consulting.html
9. http://www.conx2u.com/index.php?option=com_content&view=article&id=53&Itemid=55
10. http://en.wikipedia.org/wiki/Bulletin_board_system

For any query Feel Free to ask