CMSC 128

Introduction to Software Engineering
Second Semester AY 2009-2010

jachermocilla@uplb.edu.ph

Project Management

- Developing software requires project management
- Majority of software development projects fail because of technical problems or poor project management
- Project Management is an umbrella activity
- Effective software project management depends on People, Problem, and Process

People

- Software engineering work is an intensively human endeavor
- Players in the software process
  - Senior managers
  - Project (technical) managers
  - Practitioners
  - Customers
  - End users

Team Leaders

- Team leaders attempt to maximize the skills and abilities of each person in the development team
- How do we select a team leader?
  - Motivation
  - Organization
  - Ideas or innovation

Team Leaders

- Other traits of effective project manager
  - Problem solving – diagnosing problems and providing solution
  - Managerial identity – taking charge
  - Achievement – rewarding accomplishments
  - Influence and team building – 'reading' people

The Team

- Options for applying human resources to a project that will require N people working for K years
  - N individual to M different functional tasks
  - N individual to M different functional tasks (M < N)
  - N individuals into T teams to M different functional tasks
- Project manager is responsible for the coordination

Team Organization

- Democratic decentralized (DD)
  - No permanent leader
  - Assigns short term tasks coordinator
  - Decision made by group consensus
  - Horizontal communication
- Controlled decentralized (CD)
  - With a defined leader and secondary leaders
  - Decision made by groups consensus
  - Horizontal and vertical communication

Team Organization

- Controlled Centralized (CC)
  - Team leader manages problem solving and team coordination
  - Vertical communication between team members and team leaders

Team Organization

- Project factors that should be considered when planning team structure
  - Difficulty of problem to solve
  - Size of resultant program (LOC, FP)
  - Team lifetime
  - Modularity
  - Required quality and reliability
  - Rigidity of delivery date
  - Communication required for project

Team Organization

Type	**DD**	**CD**	**CC**
Difficulty			
High	X		
Low		X	X
Size			
Large		X	X
Small	X		
Team Lifetime			
Short		X	X
Long	X		
Modularity			
High		X	X
Low	X		
Reliability			
High	X	X	
Low			X
Delivery Date			
Strict			X
Lax	X	X	
Sociability			
High	X		
Low		X	X

Team Organization

- The objective for every project manager is to help create a team that exhibits cohesiveness, regardless of the team organization
- Team must have a common definition of success or any identifiable team spirit
- The whole is greater than the sum of the parts
- Establish momentum

Coordination and Communication

- *Scale, uncertainty,* and *interoperability* are key characteristics of modern software
- Effective methods for coordinating people who do the work must be established
  - establish mechanism for *formal(impersonal)* and *informal(personal)* communication among team members and between teams

Coordination Techniques

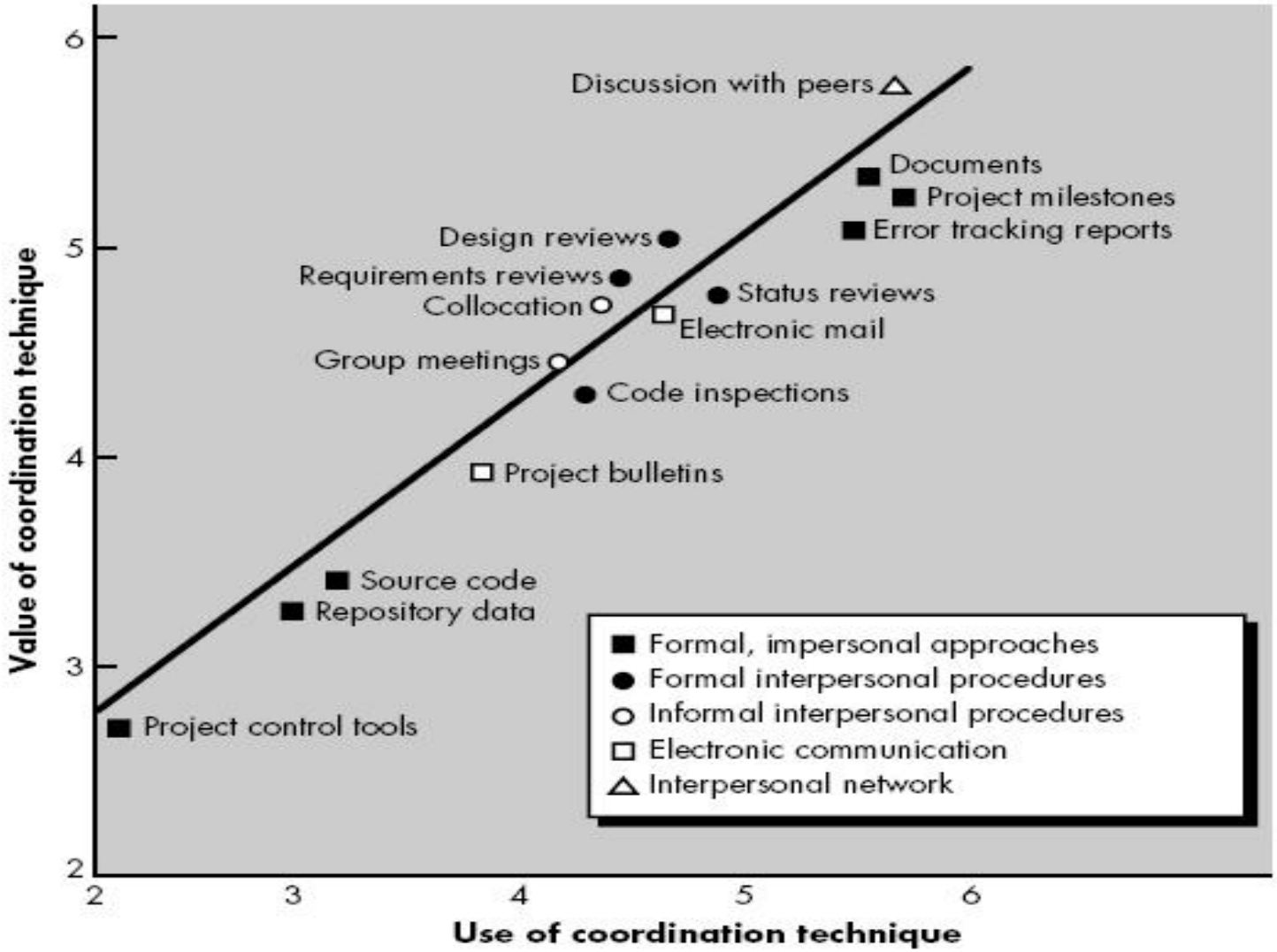
- Formal, impersonal
  - documents, deliverables, memos, schedules, change requests
- Formal, interpersonal
  - quality assurance procedures, status review meetings, design and code inspections
- Informal, interpersonal
  - information dissemination meetings, problem solving, collection of requirements

Coordination Techniques

- Electronic
  - email, electronic bulletin boards, web sites, video conferencing
- Interpersonal network
  - communication with those outside of project

Coordination Techniques

FIGURE 3.1
Value and Use of Coordination and Communication Techniques



Problem

- At the start of project, quantitative estimates and plan are required but solid information is unavailable
- Examine the problem at the very beginning of the project: *scope* must be established and bounded

Defining the Scope

- Context
  - How does the software to be built fit into a larger system or business context?What constraints are imposed as a result of the context?
- Information objectives
  - What customer-visible data objects are produced as output of software?What data objects are required as input?

Defining the Scope

- Function and performance
  - What function does the software perform to transform input data into output? Are any special performance characteristics to be addressed?

Scope

- Must be unambiguous and understandable at management and technical levels
- Must be bounded: quantitative data must be stated explicitly, constraints and limitations are noted, and mitigating factors(algorithms) are described

Problem Decomposition

- also called *partitioning,* problem is divided into small manageable subproblems
- no attempt to fully decompose problem during scoping activity, will be done in requirements analysis
- performed on the *functionality* to be delivered and the *process* that will be used to deliver the product

Process

- Generic phases apply: definition, development, and maintenance
- Which software process model to use?
  - linear sequential, prototyping, RAD, incremental, spiral, component assembly, etc
- Project manager decides which model to use and creates a preliminary plan, process decomposition begins
  - work tasks for framework activities are created

Melding Problem and Process

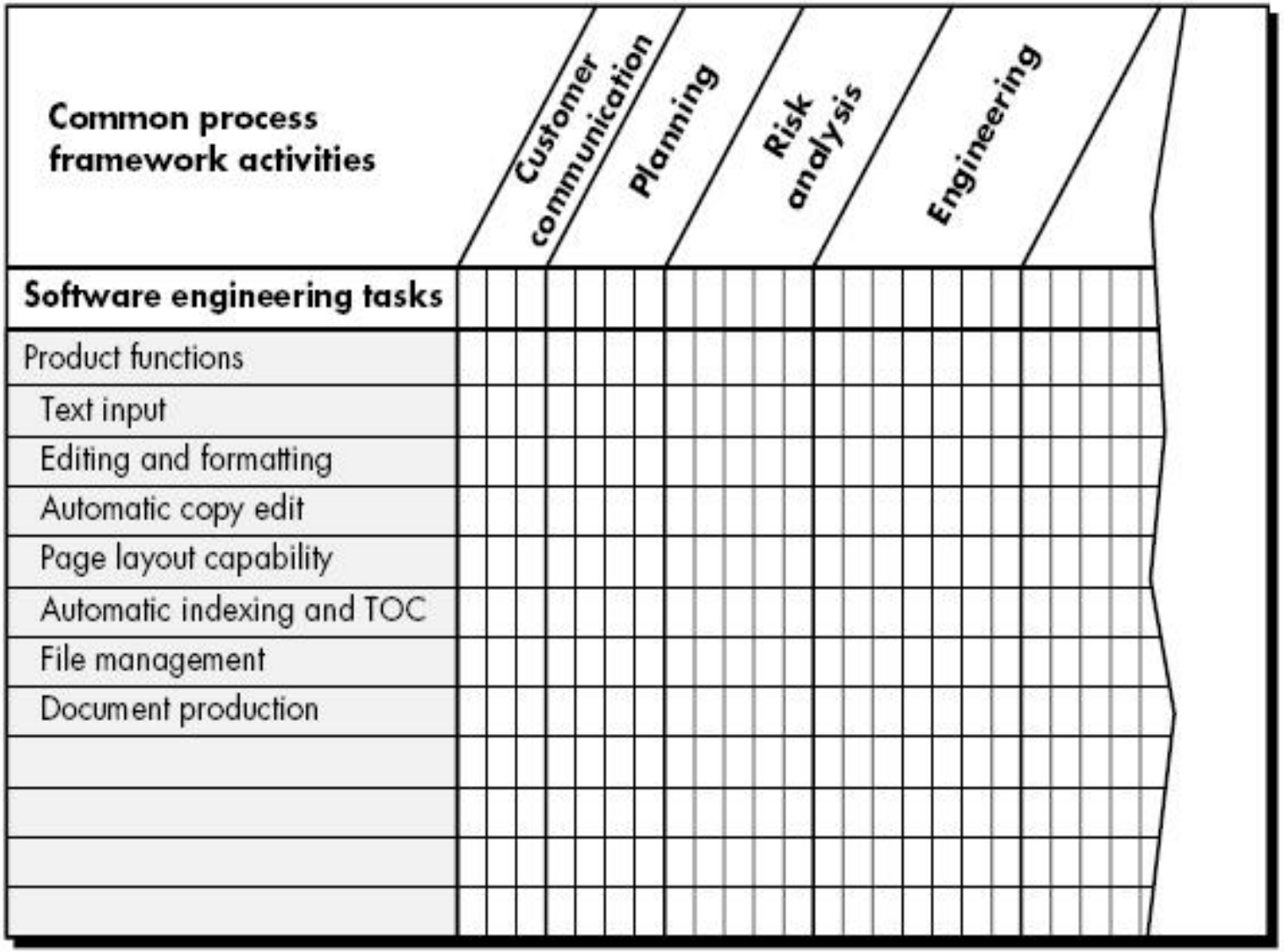
- Framework activities are identified(ex: see spiral model)
- Functions identified in the scoping activity are matched with the framework activities in a matrix
- Top row contains the framework activities, second row contains the work tasks associated for each framework activity and the leftmost column the functions

Melding Problem and Process

- The team needs to estimate resource requirements for each cell, start and end dates for each task, and work products to be produced in each cell

Melding Problem and Process

FIGURE 3.2
Melding the Problem and the Process



Process Decomposition

- Software team should have significant flexibility
  - in choosing SE paradigm(process model) that is best for the project
  - the software engineering tasks that populate the process model once it is chosen
- Examples
  - Small project-linear sequential
  - Time constrained-RAD, incremental

Process Decomposition

- Common Process Framework is adapted to selected proces model
- Framework activities can be fitted to any paradigm selected
- Selecting the work tasks for each framework activity will vary depending on the project

Process Decomposition

- Customer Communication Activity for a simple project
  1. Develop list of clarification issues
  2. Meet with the customer to address clarification issues
  3. Jointly develop a statement of scope
  4. Review the state of scope with all concerned
  5. Modify the statement of scope as required

Process Decomposition

- Customer Communication Activity for a complex project
  1. Review the customer request
  2. Plan and schedule a formal, facilitated meeting with customer
  3. Conduct research to define proposed solutions and existing approaches
  4. Prepare a 'working document' and an agenda for the formal meeting
  5. Conduct the meeting

The Project

- 90-90 Rule
  - The first 90 percent of a system absorbs 90 percent of the allotted effort and time
  - The last 10 percent takes the other 90 percent of alloted effort and time
- A project that follows this rule is in trouble!

A project in trouble

- Assessment of progress is flawed
- Cannot calibrate progress because quantitative metrics are unavailable
- Project plan was not designed to accommodate resources required at the end of the project
- Risks have not been considered explicitly
- Schedule is unrealistic and flawed

Summary

- Software project management is an umbrella activity within software engineering
- People, problem, and process has substantial influence in software project management
- Activities in project management encompass measurements and metrics, estimation, risk analysis, schedules, tracking, and control

Reference

- Roger S. Pressman.Software Engineering: A Practitioner's Approach, 4th Ed.McGraw-Hill,1997. Chapter 3