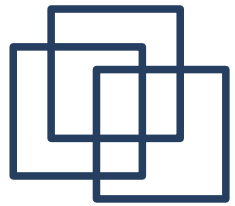


CMSC 128

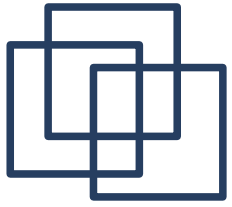
Introduction to Software Engineering Second Semester AY 2007-2008

jachermocilla@uplb.edu.ph



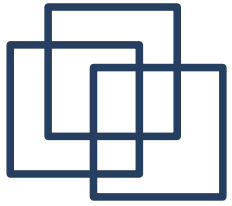
Scheduling and Tracking

- Why is software delivered late?
 - Unrealistic deadline established by someone outside the SE group
 - Changes in requirements not reflected in schedule
 - Honest underestimate of effort and resources
 - Risks not considered when project started
 - Technical and human difficulties
 - Miscommunication
 - Failure to recognize and act on possible delay
-



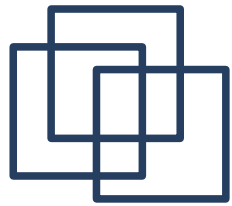
Basic Principles

- How do projects fall behind schedule?
 - 'One day at a time.' --F Brooks
- In a project, a set of small tasks must be performed to accomplish a goal
 - Some of these tasks are 'critical' and may affect schedule
- Define all tasks, identify the critical tasks, track the progress of the critical tasks



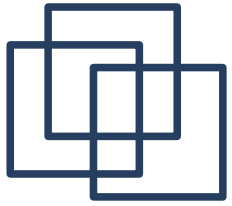
Basic Principles

- Scheduling
 - Distributes estimated effort across the planned project duration by allocating the effort to specific software engineering tasks
 - Schedules evolves over time
 - From macroscopic (framework activities) to microscopic (tasks)
 - First view: End date for release is set
 - Second view: End date is set by SE organization



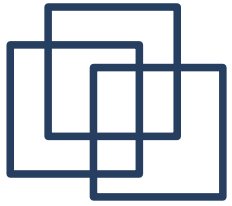
Basic Principles

- Compartmentalization
 - Product and process decomposition
- Interdependency
 - Some tasks must occur in sequence, others in parallel
- Time allocation
 - Each task must be allocated some work units
 - Start and completion date are functions of interdependency



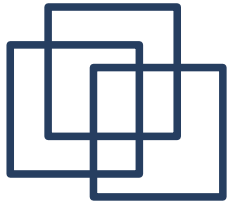
Basic Principles

- Effort Validation
 - With a defined number of staff, no more than the allocated staff should have been allocated at a given time
- Defined responsibilities
 - Every task scheduled must be assigned to a specific team member
- Defined outcomes
 - Deliverables(work product) are produced



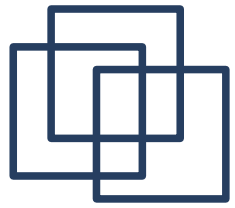
Basic Principles

- Defined milestones
 - Task or group of tasks is associated with a project milestone
 - A milestone is accomplished when one or more work products have been reviewed for quality and have been approved



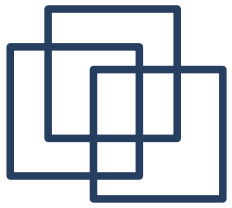
People and Effort

- In a small project, a single person can do everything: analysis, design, code, test, etc.
- Big projects require more people
- So, if project is behind schedule, perhaps the project is big! Add more people??
 - Consider communication and training



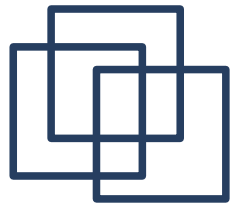
People and Effort

- Consider four developers individually capable of producing 5K LOC/yr
- Produce a team from these developers, there will be six communication paths, assume that productivity will be reduced by 250 LOC/yr.
- What will be the productivity of each developer now?
- Imagine if we add two more developers because project is behind schedule!



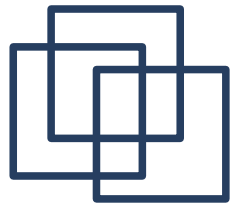
People and Effort

- Relationship between people and productivity is not linear
- Can we conclude then that forming teams is counterproductive?
 - NO!
 - If communication serves to improve quality: formal technical reviews
- Benefit can be gained by using fewer people over a somewhat longer time span



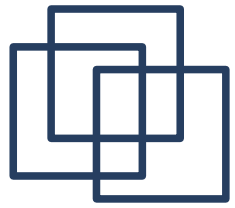
Effort Distribution

- Estimation produces estimate of effort
- 40-20-40 rule
- Characteristics of project dictate the distribution of effort
 - Project planning – 2-3 percent
 - Requirements Analysis – 10–25 percent
 - Software Design – 20-25 percent
 - Coding – 15-20 percent
 - Testing – 30-40 percent



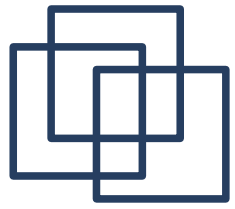
Defining Task Sets

- At the start of the project, a software process model is selected
- Process model selected consists of task sets (applies to any software process model)
- Task sets vary depending on the project
- A task set is a collection of software engineering work tasks, milestones, and deliverables that must be accomplished to complete a project



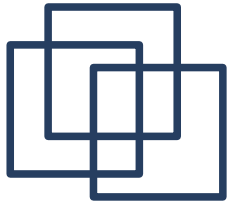
Defining Task Sets

- Task set should provide enough discipline to achieve quality and not burden the team with unnecessary work
- Task sets are designed to accommodate different types of projects and different degrees of rigor



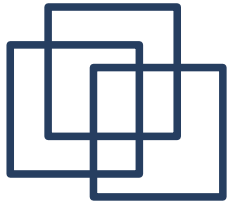
Types of Projects

- Concept Development
 - Explore some new concept, apply new tech
 - New Application Development
 - From customer request
 - Application Enhancement
 - Modifications observable by the end user
 - Application Maintenance
 - Modifications not obvious to end user
 - Reengineering - rebuilding
-



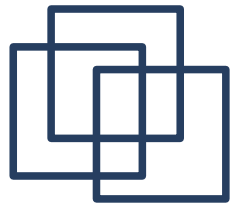
Degree of Rigor

- Casual
 - All framework activities are applied but with only a minimum task set: umbrella activities are minimized documentation requirements are reduced
- Structured
 - All framework activities are applied, including all umbrella activities
- Strict
 - Same as structured but robust documentation



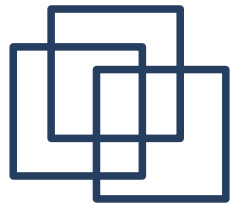
Degree of Rigor

- Quick Reaction
 - Framework activities are applied, but because of emergency situations, only essential activities will be applied
 - 'Back-filling', 'to follow'



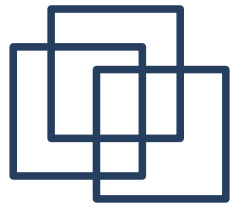
Adaptation Criteria

- Selecting the degree of rigor must be systematic, based on some criteria
 - Adaptation Criteria: size of project, number of potential users, mission criticality, application longevity, stability of requirements, ease of developer/customer communication, maturity of applicable technology, performance constraints, embedded characteristics, staffing, reengineering
-



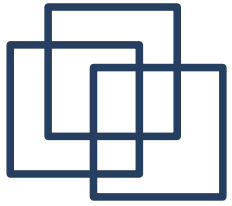
Selecting SE Tasks

- Developing project schedule requires the task set to be distributed on the project timeline
- Task set vary depending on the project type and degree of rigor
- One type of project may flow smoothly to the next: from concept to new application



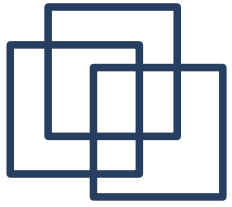
Concept Development

- Major Tasks (macroscopic)
 - Concept scoping
 - Concept planning
 - Technology risk assessment
 - Proof of concept
 - Concept implementation
 - Customer reaction to concept
- Major tasks refined through decomposition to create final schedule



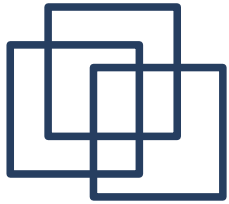
Task Networks

- Tasks and subtasks have interdependencies based on their sequence
- Some tasks can be performed in parallel
- Task network is a graphical representation of task flow
- Determine tasks that lie on the critical path
 - Tasks that determine overall schedule



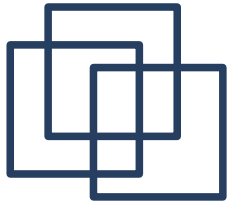
Scheduling

- Program Evaluation and Review Technique (PERT) and Critical Path Method (CPM)
 - Requires
 - Estimates of effort
 - Decomposition of product function
 - Selection of process model
 - Selection of project type and task set
- Work Breakdown Structure (WBS) – task decomposition as task network



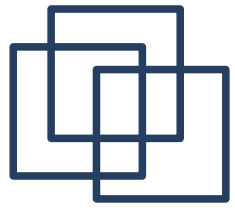
Scheduling

- What can PERT/CPM do?
 - Determine critical path
 - Chain of tasks that determine the duration
 - Establish most likely time estimates for individual tasks
 - Calculate boundary times
 - Earliest time to start
 - Latest time to start
 - Earliest finish
 - Latest finish
 - Float time (leeway)



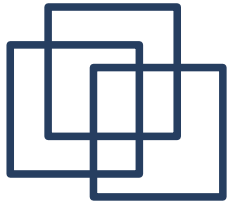
Gantt Charts

- WBS is created and entered in an automated scheduling tool
- For each task: effort, duration, start date, and human resource are specified
- Created for entire project or portions of it
- Example tools: MS Project, Imendio/Planner (linux)



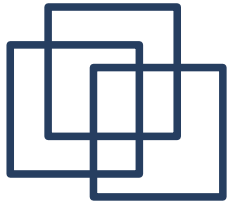
Project Tracking

- Project schedule provides roadmap
- How to track?
 - Periodic project status meeting
 - Evaluate results of all reviews
 - Determine if milestones have been achieved
 - Compare actual start date to planned start date
 - Informal meeting with practitioners to obtain subjective assessment of progress



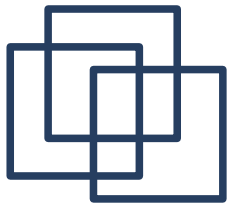
Control

- Employed to administer project resources, cope with problems, and direct project staff
- Project is going well, light control
- When problems occur
 - Allocate resource
 - Redeploy staff
 - Redefine Schedule



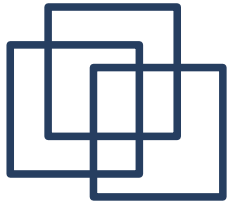
Control

- Deadline pressures??!!!
- Time-boxing
 - Complete product may not be delivered on time
 - Incremental approach is selected
 - Schedule of tasks associated with increment is adjusted backwards from deadline of increment
 - When task hits boundary, work stops and next task begins



Summary

- Scheduling is the culmination of planning activity
 - Scheduling starts with process decomposition
 - Task set is adapted based on type and degree of rigor
 - Scheduling tools include PERT, CPM, Gantt Chart, Project Table
 - Schedule used to track and control project
-



Reference

- Roger S. Pressman. Software Engineering: A Practitioner's Approach, 4th Ed. McGraw-Hill, 1997. Chapter 5