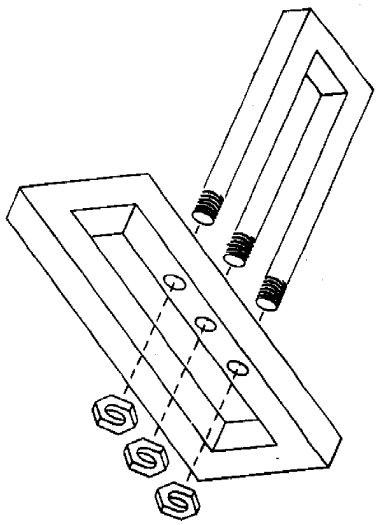
Object-Oriented Software Engineering
Using UML, Patterns, and Java

Project Management

Can you develop this?



Software Production has a Poor Track Record Example: Space Shuttle Software

- Cost: $10 Billion, millions of dollars more than planned
- Time: 3 years late
- Quality: First launch of Columbia was cancelled because of a synchronization problem with the Shuttle's 5 onboard computers.
  - **Error was traced back to a change made 2 years earlier when a programmer changed a delay factor in an interrupt handler from 50 to 80 milliseconds.**
  - **The likelihood of the error was small enough, that the error caused no harm during thousands of hours of testing.**
- Substantial errors still exist.
  - **Astronauts are supplied with a book of known software problems "Program Notes and Waivers".**

Management vs Project Mangement

- Management is usually defined in terms of functions: planning, organizing, directing, controlling and communicating
  - **Management is getting things done through people.**
- Project management is defined in the context of a project
  - **Project management tries to accomplish a specific task within a time frame and defined resources.**
- Think about this distinction as we go through the next slides

Software project management

- Concerned with activities involved in ensuring that software is delivered on time and on schedule and in accordance with the requirements of the organisations developing and procuring the software
- Project management is needed because software development is always subject to budget and schedule constraints that are set by the organisation developing the software
- Software Management Distinctions:
  - **The product is intangible**
  - **The product is uniquely flexible**
  - **Software engineering is not recognized as an engineering discipline with the same status as mechanical, electrical engineering, etc.**
  - **The software development process is not standardised**
  - **Many software projects are 'one-off' projects**

Management activities

- Proposal writing
- Project planning and scheduling
- Project costing
- Project monitoring and reviews
- Personnel selection and evaluation
- Report writing and presentations

- These activities are not peculiar to software management
  - **Many techniques of engineering project management are equally applicable to software project management**
  - **Technically complex engineering systems tend to suffer from the same problems as software systems**

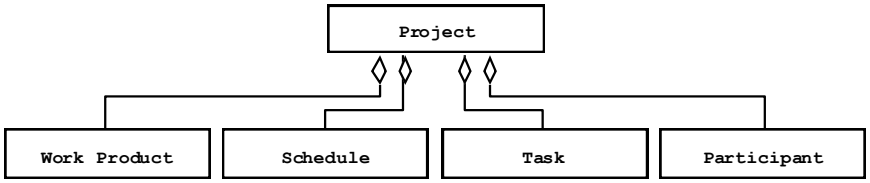
Outline of today's lecture

- Basic definitions: Project, Project Plan, Project Agreement
- Software Project Management Plan
  - **Project Organization**
  - **Managerial Processes**
  - **Technical Processes**
  - **Work Packages**
- Building a House (Example to illustrate the concepts)
- Work Breakdown Structures
- Dependency Graphs
- Tasks, Activities and Project Functions

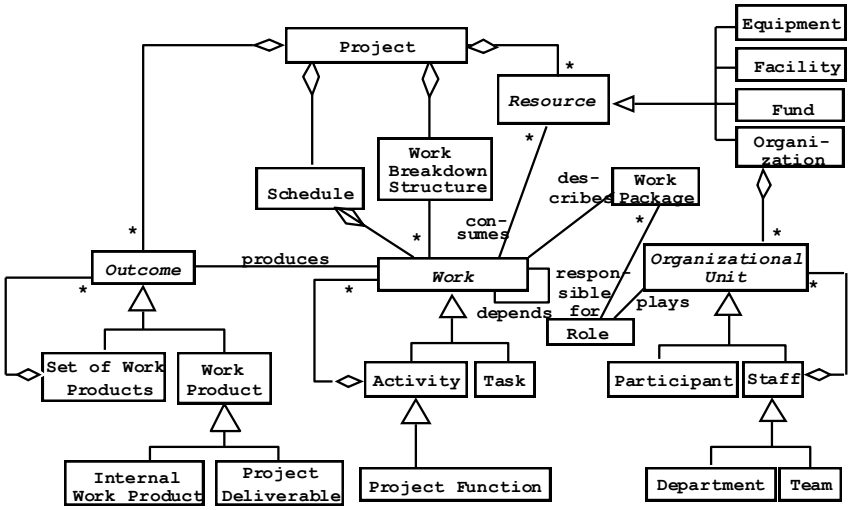
Basic Definitions: Project and Project Plan

- Software Project:
  - **All *technical* and *managerial* activities required to deliver the deliverables to the client.**
  - **A software project has a specific duration, consumes resources and produces *work products*.**
  - **Management categories to complete a software project:**
    - **Tasks, Activities, Functions**
- Software Project Management Plan:
  - **The controlling document for a software project.**
  - **Specifies the technical and managerial approaches to develop the software product.**
  - **Companion document to requirements analysis document:**
    - **Changes in either document may imply changes in the other document.**
  - **The SPMP *may* be part of the project agreement.**

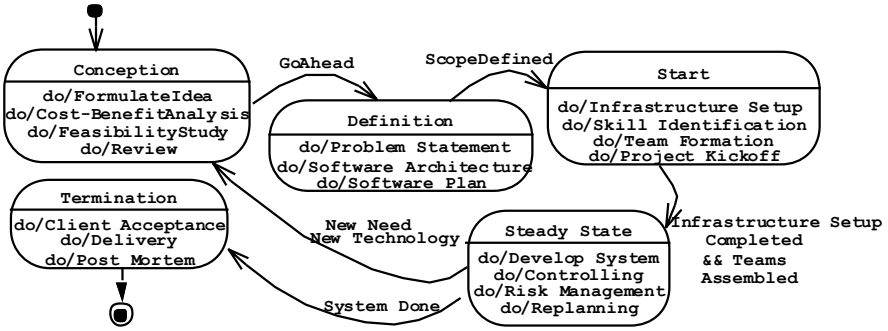
Components of a Project



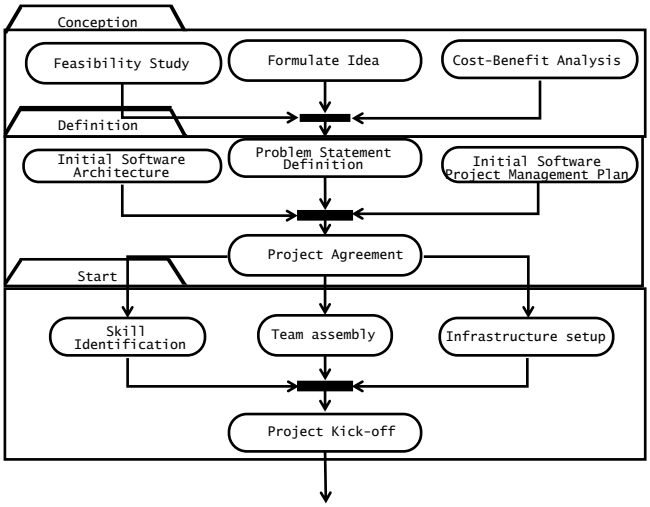
A More Complex Model



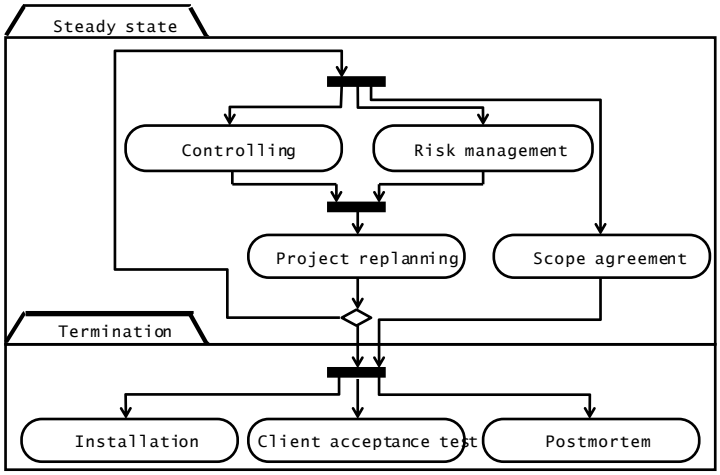
States of a Project



Management activities in a software project (continued on next slide).



Management activities in a software project (continued).



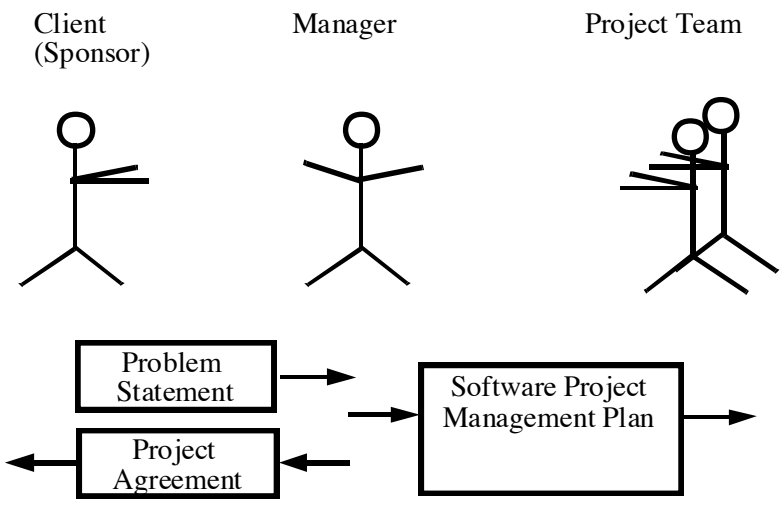
Project Agreement

- Document written for a client that defines:
  - **the scope, duration, cost and deliverables for the project.**
  - **the exact items, quantities, delivery dates, delivery location.**
- Client: Individual or organization that specifies the requirements and accepts the project deliverables.
- Can be a contract, a statement of work, a business plan, or a project charter.
- Deliverables (= Work Products that will be delivered to the client:
  - **Documents**
  - **Demonstrations of function**
  - **Demonstration of nonfunctional requirements**
  - **Demonstrations of subsystems**

IEEE Std 1058: Standard for Software Project Management Plans (SPMP)

- What it does:
  - **Specifies the format and contents of software project management plans.**
  - **It provides a standard set of abstractions for a project manager or a whole organization to build its set of practices and procedures for developing software project management plans**
  - **Abstractions: Project, Function, Activities, Tasks**
- What it does not do:
  - **It does not specify the procedures or techniques to be used in the development of the plan**
  - **It does not provide examples .**

Project Agreement,Problem Statement vs SPMP



Software Project Management Plan Template

- 0. Front Matter
- 1. Introduction
- 2. Project Organization
- 3. Managerial Process
- 4. Technical Process
- 5. Work Elements, Schedule, Budget
- Optional Inclusions

SPMP Part 0: Front Matter

- Title Page
- Revision sheet (update history)
- Preface: Scope and purpose
- Tables of contents, figures, tables

SPMP Part 1: Introduction

- 1.1 Project Overview
  - **Executive summary: description of project, product summary**
- 1.2 Project Deliverables
  - **All items to be delivered, including delivery dates and location**
- 1.3 Evolution of the SPMP
  - **Plans for anticipated and unanticipated change**
- 1.4 Reference Materials
  - **Complete list of materials referenced in SPMP**
- 1.5 Definitions and Acronyms

SPMP Part 2: Project Organization

- 2.1 Process Model
  - **Relationships among project elements**
- 2.2 Organizational Structure
  - **Internal management, organization chart**
- 2.3 Organizational Interfaces
  - **Relations with other entities (subcontractors, commercial software)**
- 2.4 Project Responsibilities
  - **Major functions and activities; nature of each; who's in charge**
  - **Matrix of project functions/activities vs responsible individuals**

Organizational Structure Example

SPMP Part 3: Managerial Process

- 3.1 Management Objectives and Priorities
  - **Describes management philosophy, priorities among requirements, schedule and budget**
- 3.2 Assumptions, Dependencies and Constraints
  - **External events the project depends on, constraints under which the project is to be conducted**
- 3.3 Risk Management
  - **Identification and assessment of risk factors, mechanism for tracking risks, implementation of contigency plans**
- 3.5 Monitoring and Controlling Mechanisms
  - **Frequency and mechanisms for reporting**
- 3.4 Staffing Plan
  - **Numbers and types of personnel required to conduct the project**

Examples of Risk Factors

- Contractual risks
  - **What do you do if the client becomes bankrupt?**
- Size of the project
  - **What do you do if you feel the project is too large?**
- Complexity of the project
  - **What do you do if the requirements are multiplying during analysis? („requirements creep“)**
- Personnel
  - **How do you hire people? Is there a danger of people leaving the project?**
- Client acceptance
  - **What do you do, if the client does not like the developed prototype?**

Project staffing

- May not be possible to appoint the ideal people to work on a project
  - **Project budget may not allow for the use of highly-paid staff**
  - **Staff with the appropriate experience may not be available**
  - **An organisation may wish to develop employee skills on a software project**
- Managers have to work within these constraints especially when there is an international shortage of skilled IT staff

Risk management

- Risk management is concerned with identifying risks and drawing up plans to minimise their effect on a project.
- A risk is a probability that some adverse circumstance will occur.
  - **Project risks affect schedule or resources**
  - **Product risks affect the quality or performance of the software being developed**
  - **Business risks affect the organisation developing or procuring the software**

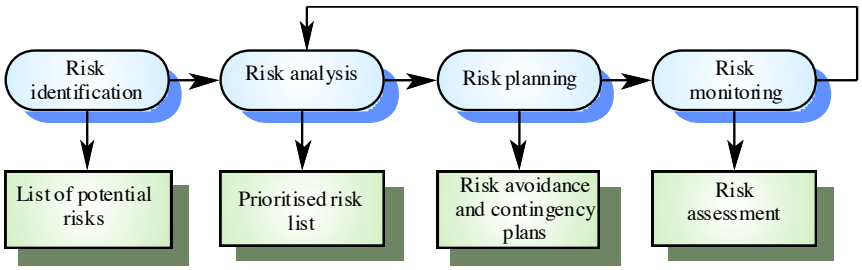
Software risks

Risk	Risk type	Description
Staff turnover	Project	Experienced staff will leave the project before it is finished.
Management change	Project	There will be a change of organisational management with different priorities.
Hardware unavailability	Project	Hardware which is essential for the project will not be delivered on schedule.
Requirements change	Project and product	There will be a larger number of changes to the requirements than anticipated.
Specification delays	Project and product	Specifications of essential interfaces are not available on schedule
Size underestimate	Project and product	The size of the system has been underestimated.
CASE tool under-performance	Product	CASE tools which support the project do not perform as anticipated
Technology change	Business	The underlying technology on which the system is built is superseded by new technology.
Product competition	Business	A competitive product is marketed before the system is completed.

The risk management process

- Risk identification
  - **Identify project, product and business risks**
- Risk analysis
  - **Assess the likelihood and consequences of these risks**
- Risk planning
  - **Draw up plans to avoid or minimise the effects of the risk**
- Risk monitoring
  - **Monitor the risks throughout the project**

The risk management process



Risk identification

- Technology risks
- People risks
- Organisational risks
- Requirements risks
- Estimation risks

Risks and risk types

Risk type	Possible risks
Technology	The database used in the system cannot process as many transactions per second as expected. Software components which should be reused contain defects which limit their functionality.
People	It is impossible to recruit staff with the skills required. Key staff are ill and unavailable at critical times. Required training for staff is not available.
Organisational	The organisation is restructured so that different management are responsible for the project. Organisational financial problems force reductions in the project budget.
Tools	The code generated by CASE tools is inefficient. CASE tools cannot be integrated.
Requirements	Changes to requirements which require major design rework are proposed. Customers fail to understand the impact of requirements changes.
Estimation	The time required to develop the software is underestimated. The rate of defect repair is underestimated. The size of the software is underestimated.

Risk analysis

- Assess probability and seriousness of each risk
- Probability may be very low, low, moderate, high or very high
- Risk effects might be catastrophic, serious, tolerable or insignificant

Risk analysis

Risk	Probability	Effects
Organisational financial problems force reductions in the project budget.	Low	Catastrophic
It is impossible to recruit staff with the skills required for the project.	High	Catastrophic
Key staff are ill at critical times in the project.	Moderate	Serious
Software components which should be reused contain defects which limit their functionality.	Moderate	Serious
Changes to requirements which require major design rework are proposed.	Moderate	Serious
The organisation is restructured so that different management are responsible for the project.	High	Serious
The database used in the system cannot process as many transactions per second as expected.	Moderate	Serious
The time required to develop the software is underestimated.	High	Serious
CASE tools cannot be integrated.	High	Tolerable
Customers fail to understand the impact of requirements changes.	Moderate	Tolerable
Required training for staff is not available.	Moderate	Tolerable
The rate of defect repair is underestimated.	Moderate	Tolerable
The size of the software is underestimated.	High	Tolerable
The code generated by CASE tools is inefficient.	Moderate	Insignificant

Risk planning

- Consider each risk and develop a strategy to manage that risk
- Avoidance strategies
  - **The probability that the risk will arise is reduced**
- Minimisation strategies
  - **The impact of the risk on the project or product will be reduced**
- Contingency plans
  - **If the risk arises, contingency plans are plans to deal with that risk**

Risk management strategies

Risk	Strategy
Organisational financial problems	Prepare a briefing document for senior management showing how the project is making a very important contribution to the goals of the business.
Recruitment problems	Alert customer of potential difficulties and the possibility of delays, investigate buying-in components.
Staff illness	Reorganise team so that there is more overlap of work and people therefore understand each other's jobs.
Defective components	Replace potentially defective components with bought-in components of known reliability.
Requirements changes	Derive traceability information to assess requirements change impact, maximise information hiding in the design.
Organisational restructuring	Prepare a briefing document for senior management showing how the project is making a very important contribution to the goals of the business.
Database performance	Investigate the possibility of buying a higher-performance database.
Underestimated development time	Investigate buying in components, investigate use of a program generator.

Risk monitoring

- Assess each identified risks regularly to decide whether or not it is becoming less or more probable
- Also assess whether the effects of the risk have changed
- Each key risk should be discussed at management progress meetings

Risk factors

Risk type	Potential indicators
Technology	Late delivery of hardware or support software, many reported technology problems
People	Poor staff morale, poor relationships amongst team member, job availability
Organisational	organisational gossip, lack of action by senior management
Tools	reluctance by team members to use tools, complaints about CASE tools, demands for higher-powered workstations
Requirements	many requirements change requests, customer complaints
Estimation	failure to meet agreed schedule, failure to clear reported defects

SPMP Part 4: Technical Process

- 4.1 Methods, Tools and Techniques
  - **Specify the methods, tools and techniques to be used on the project**
- 4.2 Software Documentation
  - **Describe the documentation plan**
- 4.3 Project Support Functions
  - **Plans for (at least) the following project support functions.**
    - **Plan to ensure quality assurance**
    - **Configuration management plan (IEEE Std 1042)**
    - **Verification and validation plan**
  - **The plans can be included in this section or there is a reference to a separate document**

SPMP Part 5: Description of Work Packages

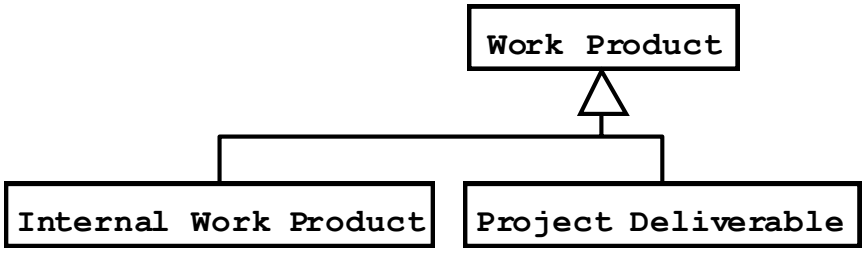
- Work Breakdown Structure (WBS) (Section 5.1)
  - **Hierarchical decomposition of the project into activities and tasks**
- Dependencies between tasks (Section 5.2)
  - **An important temporal relation: "must be preceded by"**
  - **Dependency graphs show temporal dependencies of the activities:**

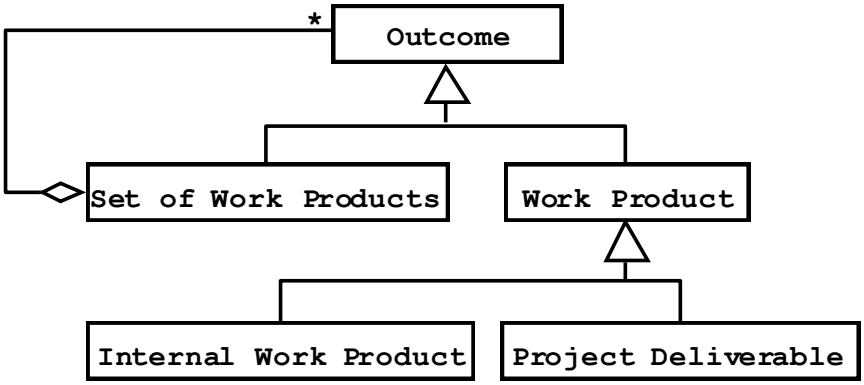
Work package vs Work product

- **Definitions from the IEEE Standard**
- **Work Package:**
  - **A specification for the work to be accomplished in an activity or task**
- **Work Product:**
  - **Any tangible item that results from a project function, activity or task.**
- **Project Baseline**:
  - **A work product that has been formally reviewed and agreed upon.**
  - **A project baseline can only be changed through a formal change procedure**
- **Project Deliverable:**
  - **A work product to be delivered to the client**

How can we model a work product?

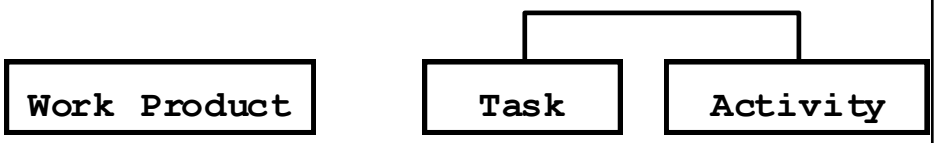
- What is a tangible item?
  - **Requirements Analysis Document**
  - **Software Project Management Plan**
  - **Agenda, Minutes**
  - **How do we model Tangible Items?**





Work products are related to Activities

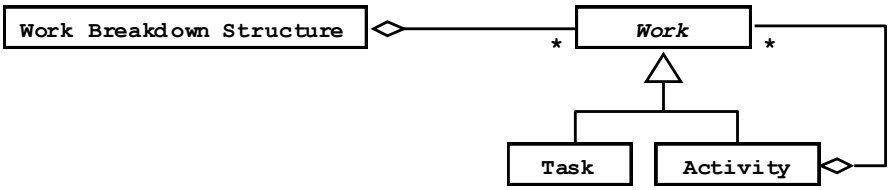
- **How do we model this?**



Work Breakdown Structure

- The hierarchical representation of all the tasks in a project is called the work breakdown structure (WBS). First Version of a UML Model

- But Tasks are Parts of Activities. What would be a better model?



Creating Work Breakdown Structures

- Two major philosophies
  - **Activity-oriented decomposition („Functional decomposition“)**
    - **Write the book**
    - **Get it reviewed**
    - **Do the suggested changes**
    - **Get it published**
  - **Result-oriented („Object-oriented decomposition“)**
    - **Chapter 1**
    - **Chapter 2**
    - **Chapter 3**
- Which one is best for managing? Depends on project type:
  - **Development of a prototype**
  - **Development of a product**
  - **Project team consist of many unexperienced beginners**
  - **Project team has many experienced developers**

Estimates for establishing WBS

- Establishing an WBS in terms of percentage of total effort:
  - **Small project (7 person-month): at least 7% or 0.5 PM**
  - **Medium project (300 person-month): at least 1% or 3 PMs**
  - **Large project (7000 person-month): at least 0.2 % or 15 PMs**
  - **(From Barry Boehm, Software Economics)**

Example: Let's Build a House

- What are the activities that are needed to build a house?

Typical activities when building a house

- Surveying
- Excavation
- Request Permits
- Buy Material
- Lay foundation
- Build Outside Wall
- Install Exterior Plumbing
- Install Exterior Electrical
- Install Interior Plumbing
- Install Interior Electrical
- Install Wallboard
- Paint Interior
- Install Interior Doors
- Install Floor
- Install Roof
- Install Exterior Doors
- Paint Exterior
- Install Exterior Siding
- Buy Pizza

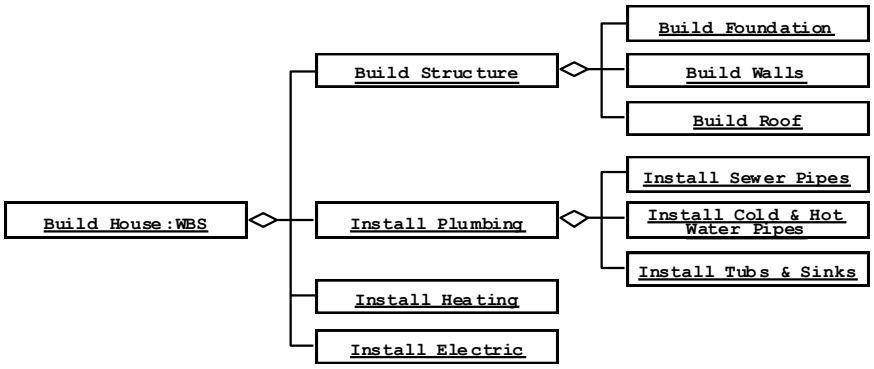
Finding these activities is a brainstorming activity.
It requires similar activities used during requirements engineering and analysis (use case modeling)

Hierarchical organization of the activities

- Building the house consists of
  - **Prepare the building site**
  - **Building the Exterior**
  - **Building the Interior**
- Preparing the building site consists of
  - **Surveying**
  - **Excavation**
  - **Buying of material**
  - **Laying of the foundation**
  - **Requesting permits**

Finding this organization involves categorization and refinement. Good after brainstorming, not during brainstorming

Partial Work Breakdown Structure



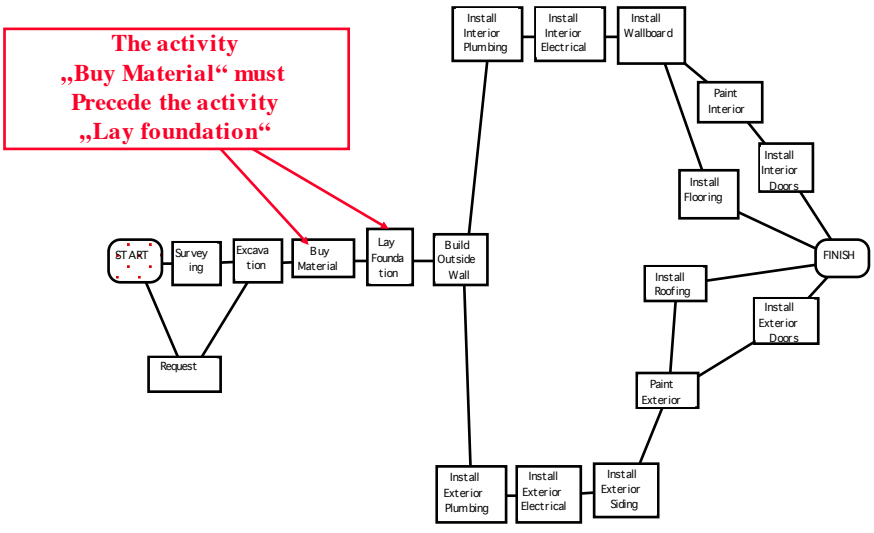
From the WBS to the Dependency Graph

- **The work breakdown structure does not show any temporal dependence among the activities/tasks**
  - Can we excavate before getting the permit?
  - How much time does the whole project need if I know the individual times?
  - What can be done in parallel?
    - **Organize tasks concurrently to make optimal use of workforce**
    - **Minimize task dependencies to avoid delays caused by one task waiting for another to complete**
  - Are there any critical actitivites, that can slow down the project significantly?
- **Temporal dependencies are shown in the dependency graph**
  - **Nodes are activities**
  - **Lines represent temporal dependencies**

Why Dependency Diagrams?

- Example:
  - **You are assigned a project consisting of 10 activities which take one week each to be completed.**
  - **How long does the project take?**
- When projects have more than 15-20 activities, one cannot to compute the schedule in the head any more.
- Dependency Diagrams are a formal notation to help in the construction and analysis of *complex schedules*

Building a House (Dependency Graph)

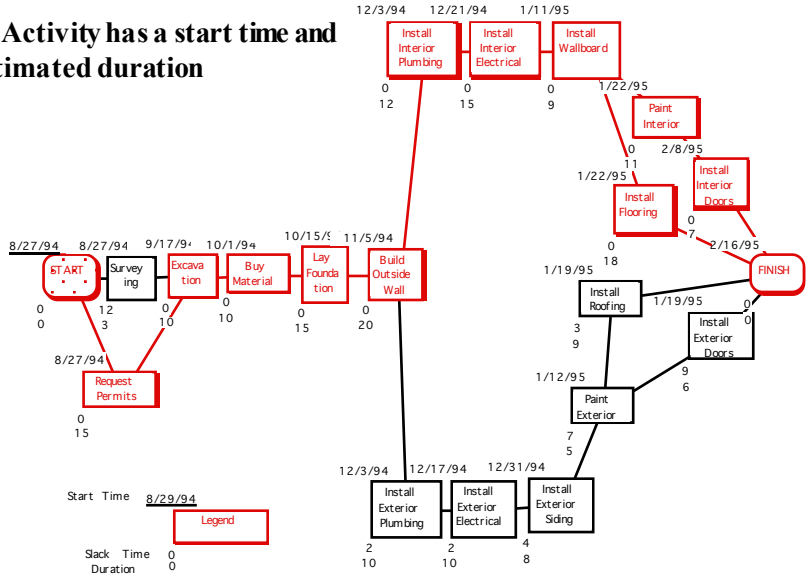


SPMP Part 5: Description of Work Packages ctd

- Resource Requirements (5.3)
  - **Estimates of the total resource (Personnel, Computer Time, Support Software) required to complete the project**
  - **Numbers and types of personnel**
  - **Computer time**
  - **Office and laboratory facilities**
  - **Travel**
  - **Maintenance and training requirements**
- Budget (5.4)
- Schedule (Section 5.5)
  - **Estimate the duration of each task**
  - **Label dependency graph with the estimates**

Building a House (PERT Chart)

Each Activity has a start time and an estimated duration



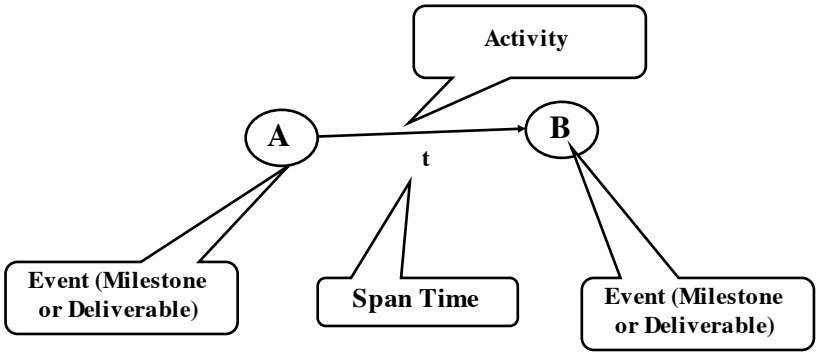
Goals of PERT Charts

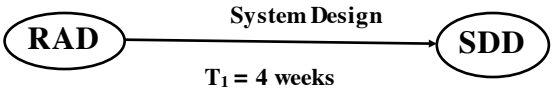
- Determination of total project time („project duration“)
- Determination of the critical path
- Determination of slack times

Dependency Diagrams

- Dependency diagrams consist of 3 elements
- **Event** (also called **milestone**): A significant occurrence in the life of a project.
- **Activity**: Work required to move from one event to the next.
- **Span time** (also called **duration** or **elapsed time):** The actual calendar time required to complete an activity.
  - Span time parameters: people's availability, parallelizability of the activity, availability of nonpersonnel resources, ….
- **Dependency Diagram** are drawn as a connected graph of nodes and arrows.
- 2 commonly used diagram notations to display dependency diagrams:
  - **1) Activity-on-the-arrow**
  - **2) Activity-in-the-node**

1) Activity-on-the-arrow Diagram Notation



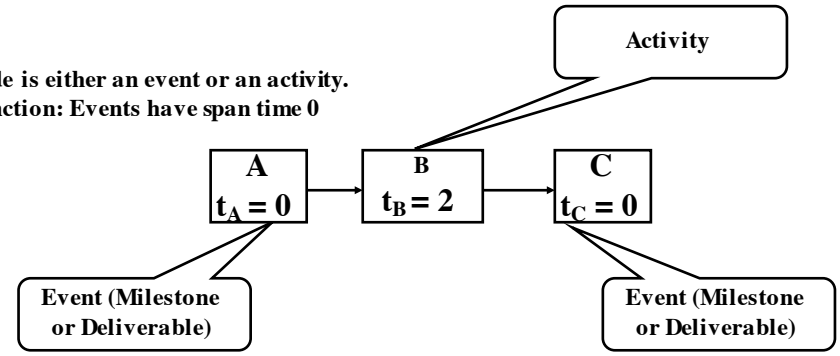


PERT

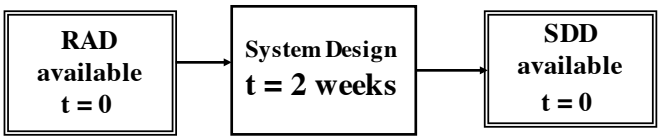
- PERT is an activity-on-the-arrow notation
- PERT = Program Evaluation and Review Technique
- Developed in the 50s to plan the Polaris weapon system in the USA.
- PERT allows to assign optimistic, pessimistic and most likely estimates for the span times of each activity.
- You can then compute the probability to determine the likelihood that overall project duration will fall within specified limits.

2) Activity-in-the-node Diagram Notation

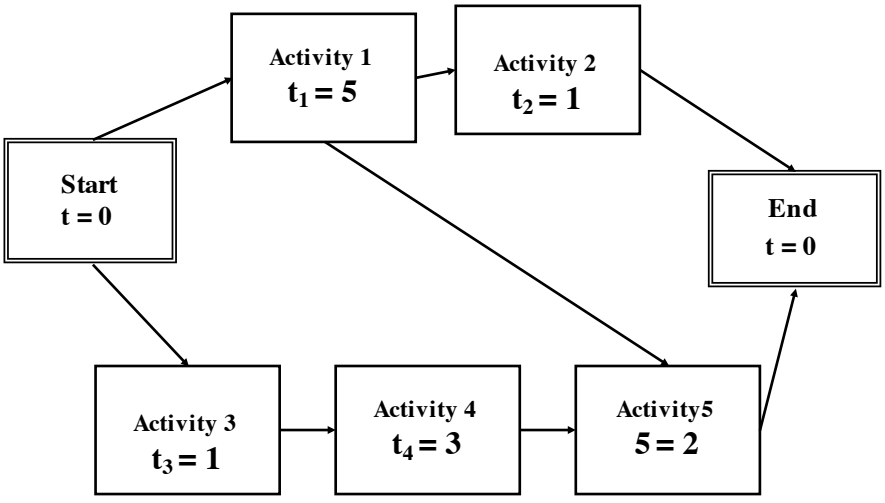
A Node is either an event or an activity.
Distinction: Events have span time 0



Milestone boxes are often highlighted by double-lines



Example of an Activity-in -the -Node Diagram



What do we do with these diagrams?

- Compute the project duration
- Determine activities that are critical to ensure a timely delivery

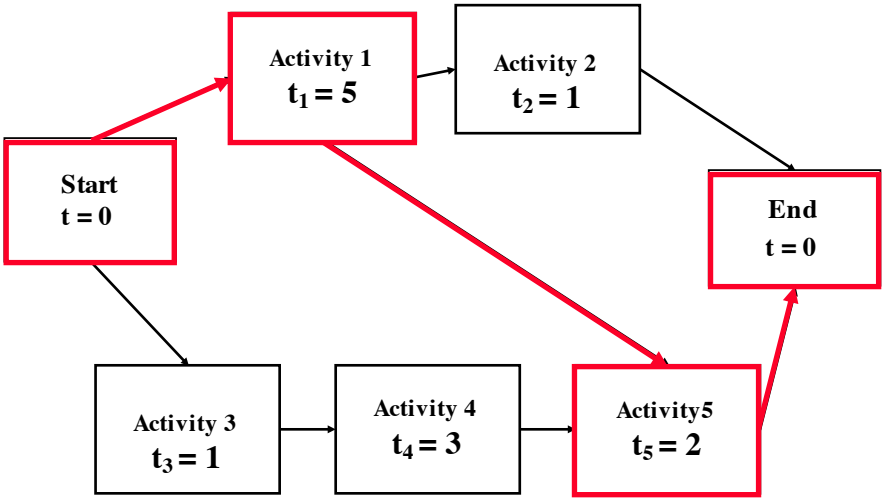
- Analyse the diagrams
  - **to find ways to shorten the project duration**
  - **To find ways to do activities in parallel**

- 2 techniques are used
  - **Forward pass (determine critical paths)**
  - **Backward pass (determine slack time)**

Definitions: Critical Path and Slack Time

- Critical path:
  - **A sequence of activities that take the longest time to complete**
  - **The length of the critical path(s) defines how long your project will take to complete.**
- Noncritical path:
  - **A sequence of activities that you can delay and still finish the project in the shortest time possible.**
- Slack time:
  - **The maximum amount of time that you can delay an activity and still finish your project in the shortest time possible.**

Example of a critical path

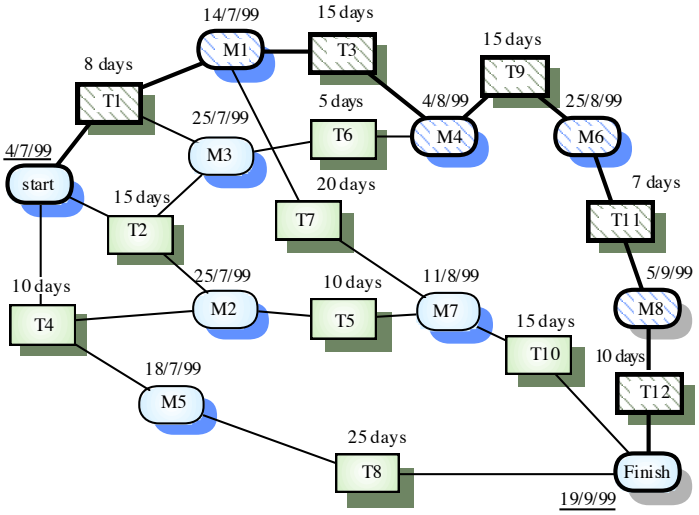


Critical path in bold face

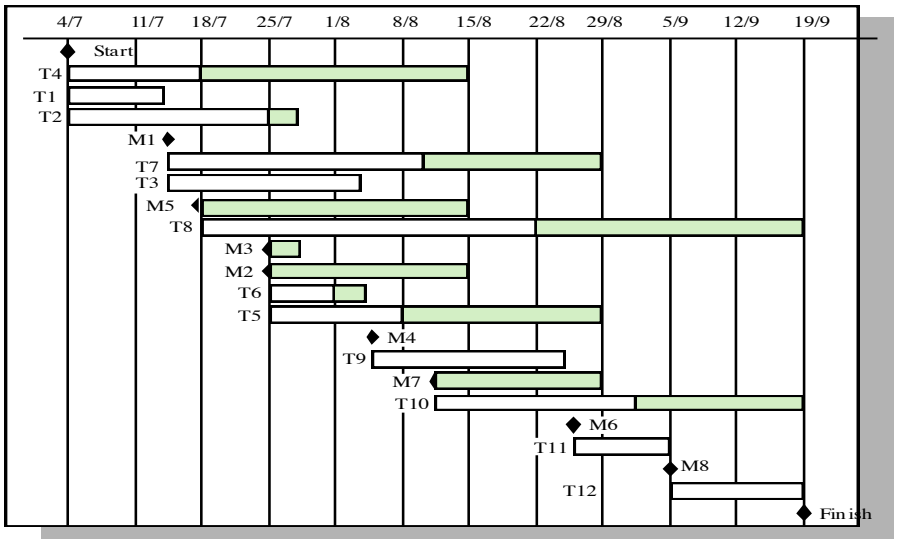
Task durations and dependencies

Task	Duration (days)	Dependencies
T1	8	
T2	15	
T3	15	T1 (M1)
T4	10	
T5	10	T2,T4 (M2)
T6	5	T1,T2 (M3)
T7	20	T1 (M1)
T8	25	T4 (M5)
T9	15	T3,T6 (M4)
T10	15	T5,T7 (M7)
T11	7	T9 (M6)
T12	10	T11 (M8)

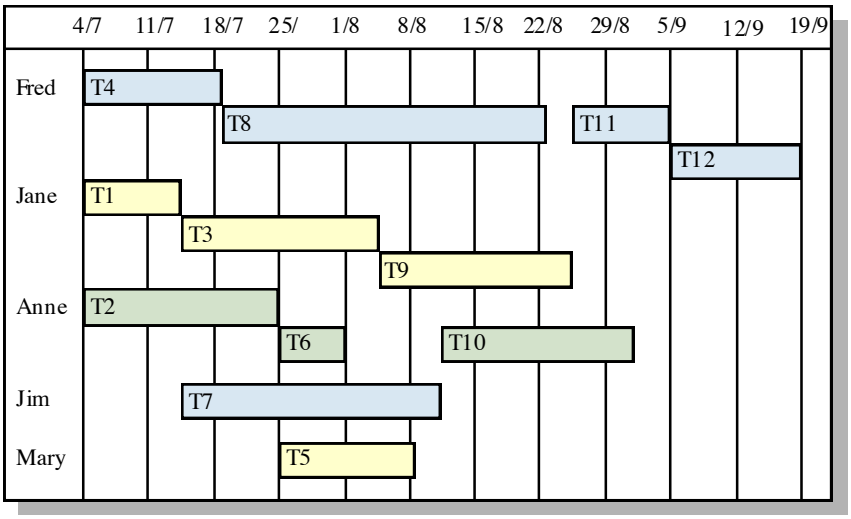
Activity network (PERT)



Activity timeline (GANTT)

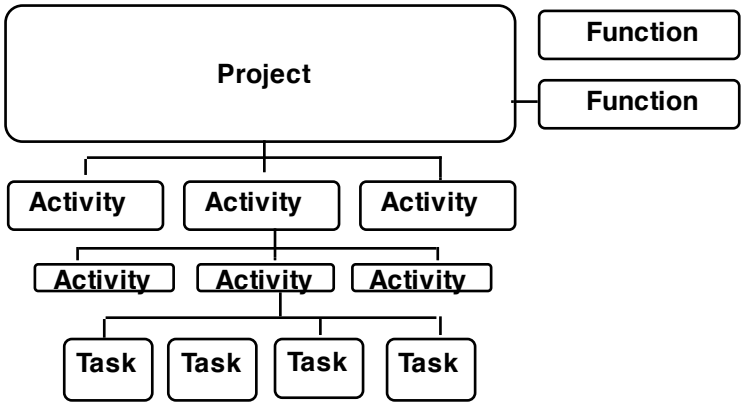


Staff allocation



Project: Functions, Activities and Tasks

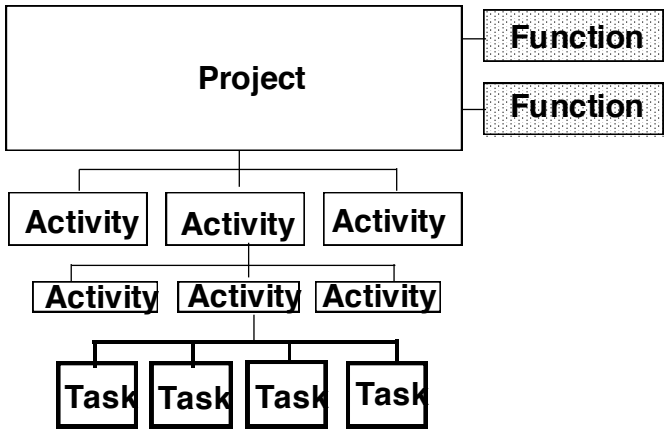
A Project has a duration and consists of functions, activities and tasks



Exercise: Write the corresponding UML class diagram!

Functions

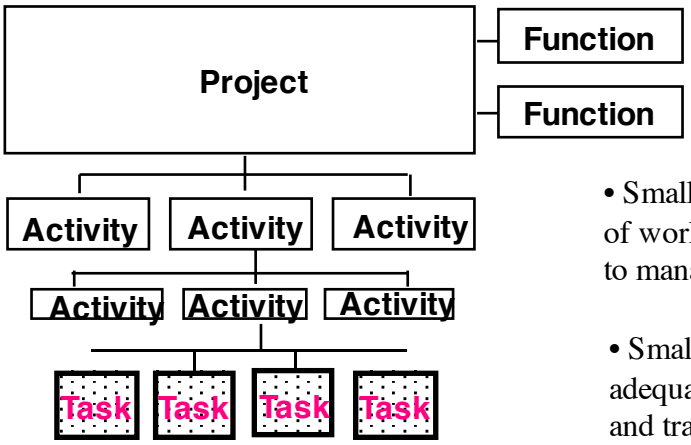
- **Definition (Project) Function**: An activity or set of activities that span the duration of the project



Functions

- Examples:
  - **Project management**
  - **Configuration Management**
  - **Documentation**
  - **Quality Control (Verification and validation)**
  - **Training**
- Question: Is system integration a project function?
  - **It Depends…**
- Mapping of terms: Project Functions in the IEEE 1058 standard are called **Integral processes** in the IEEE 1074 standard. Sometimes also called cross-development processes
- Also refer to standard ISO-IEC & IEEE 12207-2008

Tasks



- Smallest unit of work subject to management
- Small enough for adequate planning and tracking
- Large enough to avoid micro management

Tasks

- Smallest unit of management accountability
  - **Atomic unit of planning and tracking**
  - **Tasks have finite duration, need resources, produce tangible result (documents, code)**
- Specification of a task: Work package
  - **Name, description of work to be done**
  - **Preconditions for starting, duration, required resources**
    - **Other Work packages that need to be completed before this task can be started.**
  - **Work product to be produced, acceptance criteria for it**
  - **Risk involved**
- Completion criteria
  - **Includes the acceptance criteria for the work products (deliverables) produced by the task.**

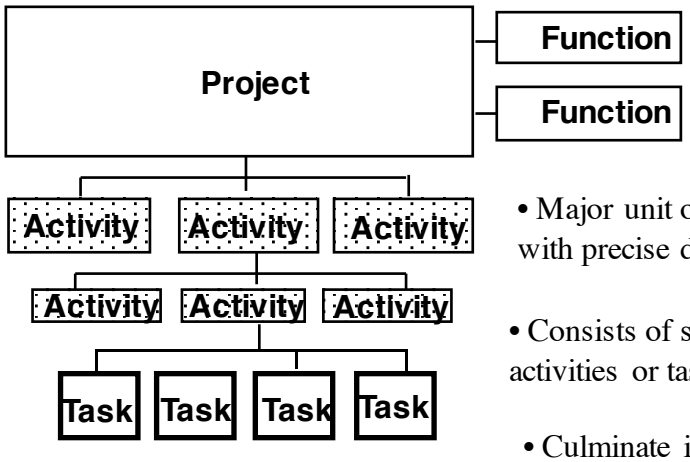
Determining Task Sizes

- Finding the appropriate task size is problematic
  - **Todo lists from previous projects**
  - **During initial planning a task is necessarily large**
  - **You may not know how to decompose the problem into tasks at first**
  - **Each software development activitity identifies more tasks and modifies existing ones**
- Tasks must be decomposed into sizes that allow monitoring
  - **Depends on nature of work and how well task is understood.**
  - **Work package usually corresponds to well defined work assignment for one worker for a week or a month.**
  - **Work assignments are also called action items**

Action Item

- **Definition Action Item**: A task assigned to a project participant
  - **What?, Who?, When?**
  - **Heuristics for Duration: be done within two weeks or a week**
- Action items should be tracked by the project manager
- They should appear on the meeting agenda in the Status Section
- Examples of Todos:
  - **Unit test class Foo**
  - **Develop project plan.**
- Example of action items:
  - **Bob posts the next agenda for the context team meeting before Sep 10, 12 noon.**
  - **The test team develops the test plan by Sep 18**

Activities

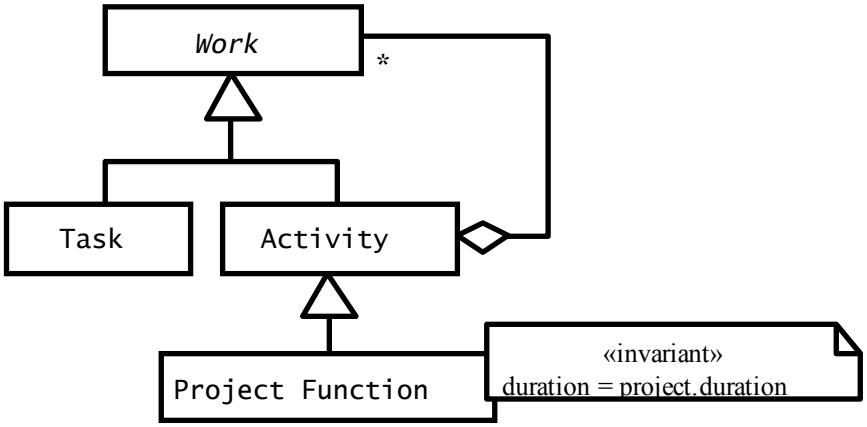


- Major unit of work with precise dates
- Consists of smaller activities or tasks
- Culminate in project milestones.

Activities

- Major unit of work
- Culminates in major project milestone:
  - **Internal checkpoint should not be externally visible**
  - **Scheduled event used to measure progress**
- Milestone often produces project baselines:
  - **formally reviewed work product**
  - **under change control (change requires formal procedures)**
- Activitites may be grouped into larger activities:
  - **Establishes hierarchical structure for project (phase, step, ...)**
  - **Allows separation of concerns**
  - **Precedence relations often exist among activities**

UML Model of Tasks, Activities and Project Functions



"Laws" of Project Management

- Projects progress quickly until they are 90% complete. Then they remain at 90% complete forever.
- When things are going well, something will go wrong.
- When things just can't get worse, they will.
- When things appear to be going better, you have overlooked something.
- If project content is allowed to change freely, the rate of change will exceed the rate of progress.
- Project teams detest progress reporting because it manifests their lack of progress.

Scheduling problems

- Estimating the difficulty of problems and hence the cost of developing a solution is hard
- Productivity is not proportional to the number of people working on a task
- Adding people to a late project makes it later because of communication overheads
- The unexpected always happens. Always allow contingency in planning

Summary

- Software engineering is a problem solving activity
  - **Developing quality software for a complex problem within a limited time while things are changing**
  - **Risk Management**
- The system models addresses the technical aspects:
  - **Object model, functional model, dynamic model**
- Other models address the management aspects
  - **WBS, Schedule are examples**
  - **Task models, Issue models, Cost models**
- Introduction of some technical terms
  - **Project, Activity, Function, Task, WBS**