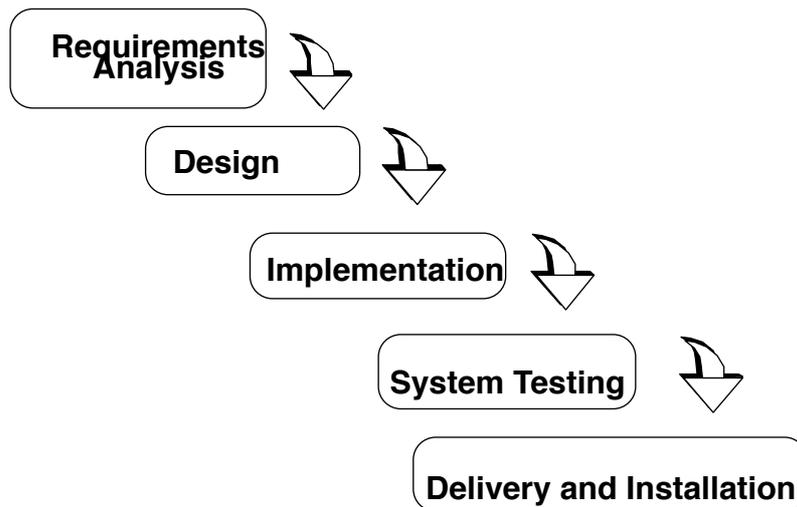
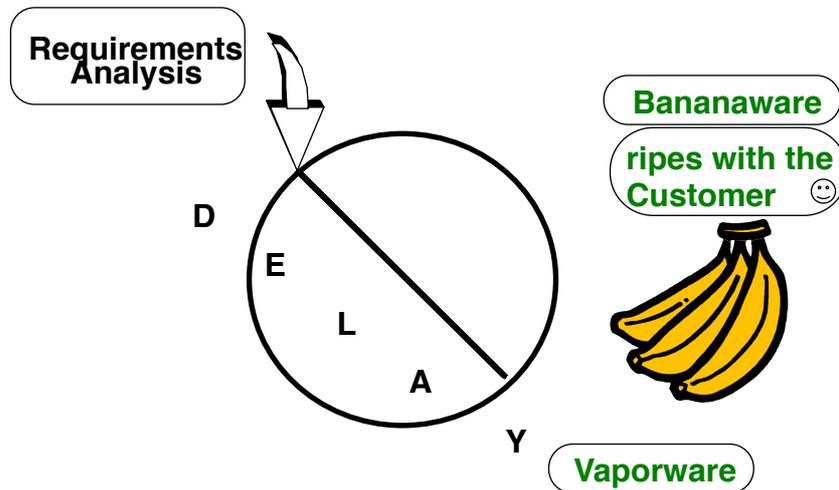


Chapter 3, Project Organization and Communication, Part 1

How it should go



How it often goes



Laws of Project Management

- Projects progress quickly until they are 90% complete
 - Then they remain at 90% complete forever
- 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
- Murphy's law:
 - "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."

Lecture Outline

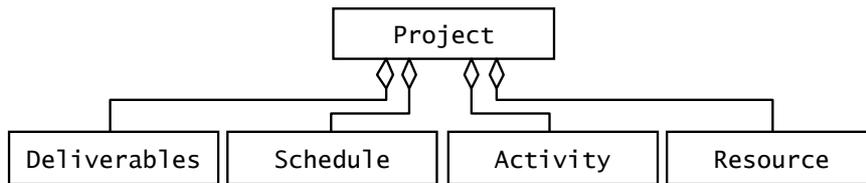
- Project Definition
- Project Organization
- Roles
- Tasks & Activities
- Work Product & Deliverables

- Focus of this lecture
 - Understand project management concepts from the developer's perspective

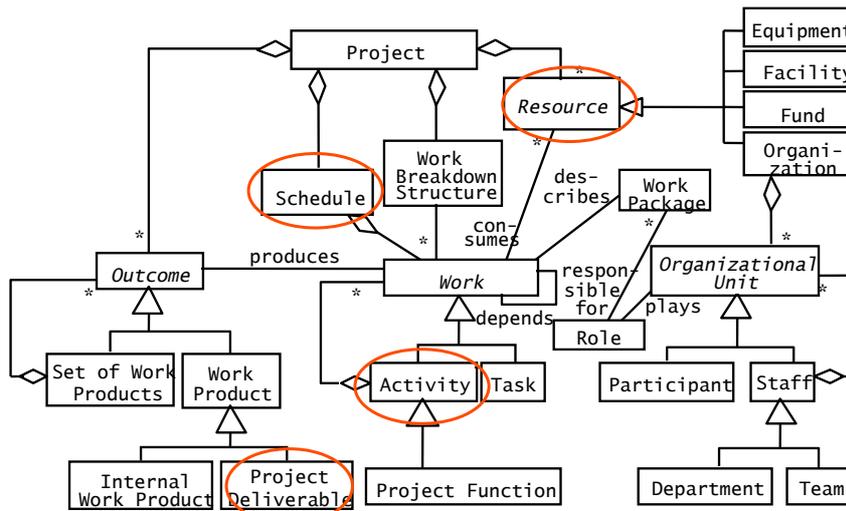
Project Definition

- A **project** is an undertaking, limited in time, to achieve a set of goals that require a concerted effort
- **A project includes**
 - A set of deliverables to a client
 - A schedule
 - Technical and managerial activities required to produce and deliver the deliverables
 - Resources consumed by the activities (people, budget)
- Focus of **project management**
 - Administer the resources
 - Maintain accountability
 - React to change
 - Make sure, the goals are met.

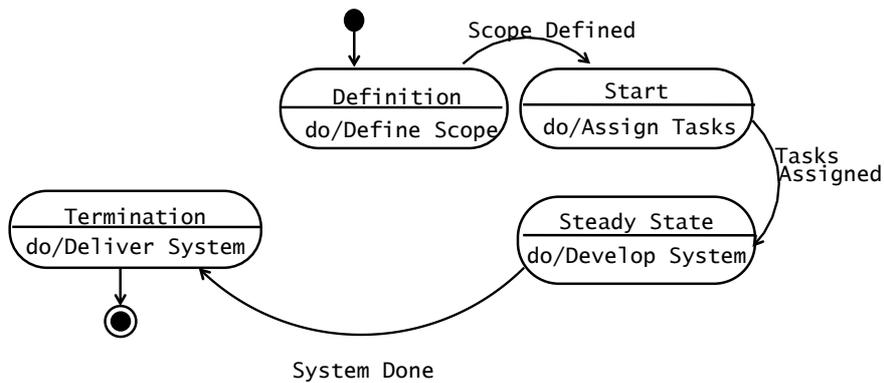
Simple Object Model of a Project



Refinement of the Model

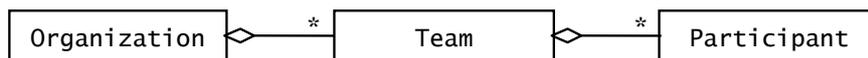


Dynamic Model of a Project

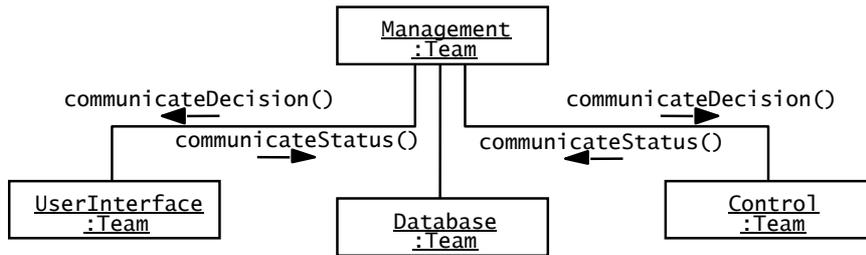


Project Organization

- A **project organization** defines the relationships among resources, in particular the participants, in a project
- A project organization should define
 - Who decides (**decision structure**)
 - Who reports their status to whom (**reporting structure**)
 - Who communicates with whom (**communication structure**)



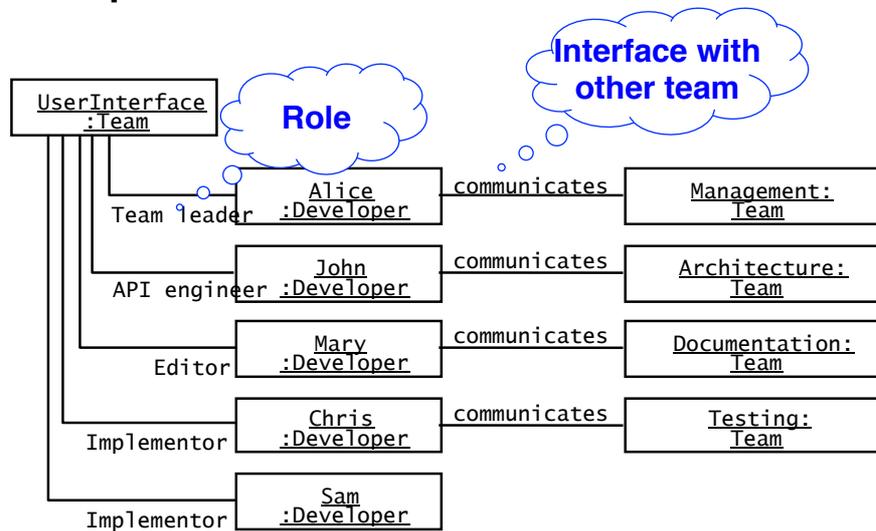
Example of a Communication Structure



Reporting vs. Communication

- Reporting supports project management in tracking project status
 - What work has been completed?
 - What work is behind schedule?
 - What issues threaten project progress?
- Reporting along the hierarchy is not sufficient when two teams need to communicate
 - A communication structure is needed
 - A participant from each team is responsible for facilitating communication between both teams
 - Such participants are called **liaison**

Example of a Communication Structure

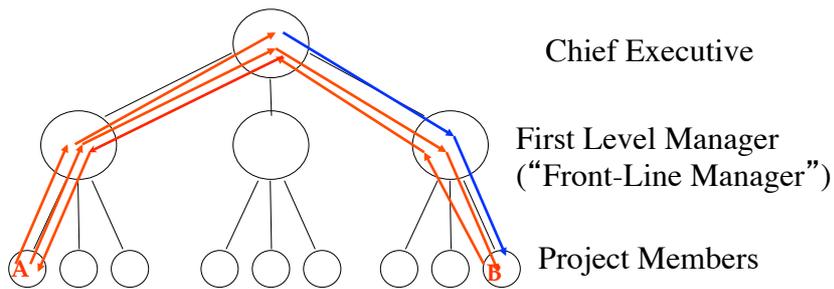


Bernd Bruegge & Allen H. Dutoit

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

13

Hierarchical Project Organization



A wants to talk to B: Information Flow

A wants to make sure B does a certain change: Controlflow

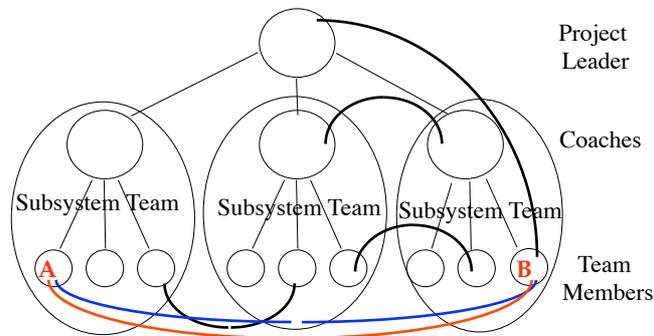
Basis of organization:
Complicated information and control flow
across hierarchical boundaries

Bernd Bruegge & Allen H. Dutoit

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

14

Peer-To-Peer Communication



A wants to talk to B: Simple Information Flow

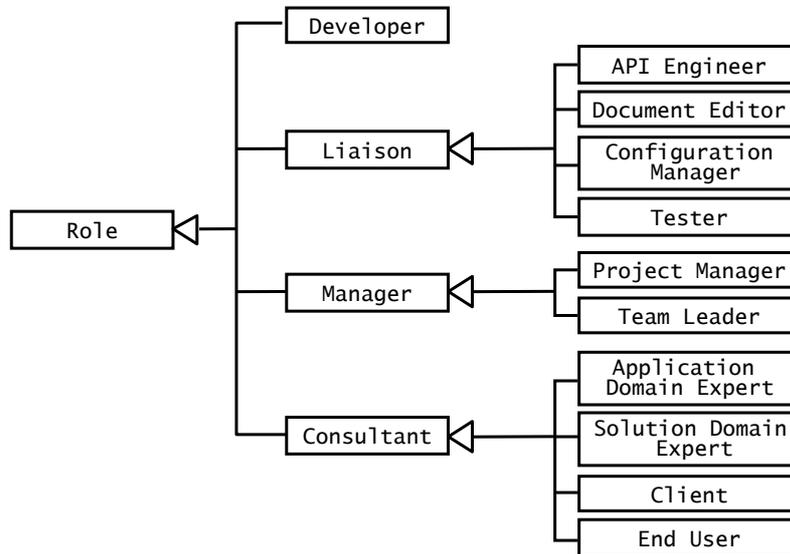
A wants to make sure B does a certain change: Simple Controlflow

Basis of organization:
Nonlinear information flow across dynamically formed units

Role

- A **role** defines a set **responsibilities** (“to-dos”)
- Examples
- **Role: Tester**
 - Write tests
 - Report failures
 - Check if bug fixes address a specific failure
- **Role: System architect**
 - Ensure consistency in design decisions and define subsystem interfaces
 - Formulate system integration strategy
- **Role: Liaison**
 - Facilitate communication between two teams.

Types of Roles in Software Organizations



Bernd Bruegge & Allen H. Dutoit

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

17

Responsibilities are assigned to Roles, Roles are assigned to People

“To Do” List for the Project

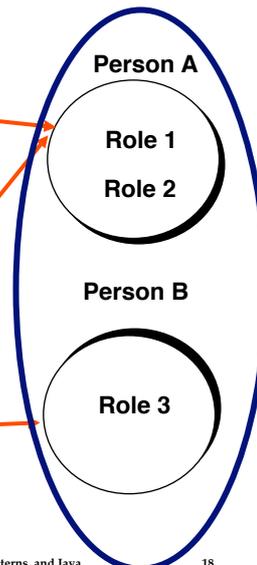
- Item 1
- Item 2
- Item 3
- Item 4
- Item 5
- Item 6
- Item 7
- Item 8
- Item 9

Role 1
Item 1
Item 2
Item 9

Role 2
Item 4
Item 5
Item 7

Role 3
Item 3
Item 6
Item 8

Team A .



Bernd Bruegge & Allen H. Dutoit

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

18

Possible Mappings of Roles to Participants

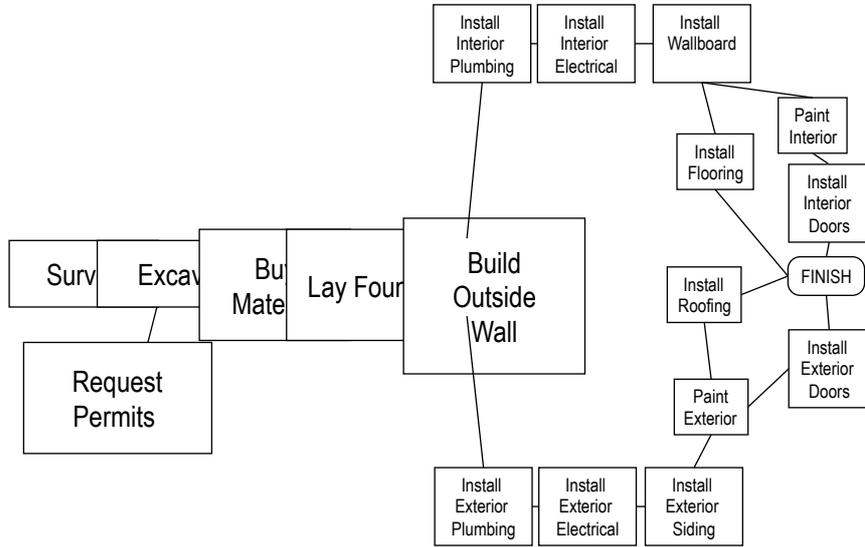
- **One-to-One**
 - Ideal but rare
- **Many-to-Few**
 - Each project member assumes several "hats"
 - Danger of over-commitment
 - Need for load balancing
- **Many-to-"Too-Many"**
 - Some people don't have significant roles
 - Lack of accountability
 - Loosing touch with project

Task

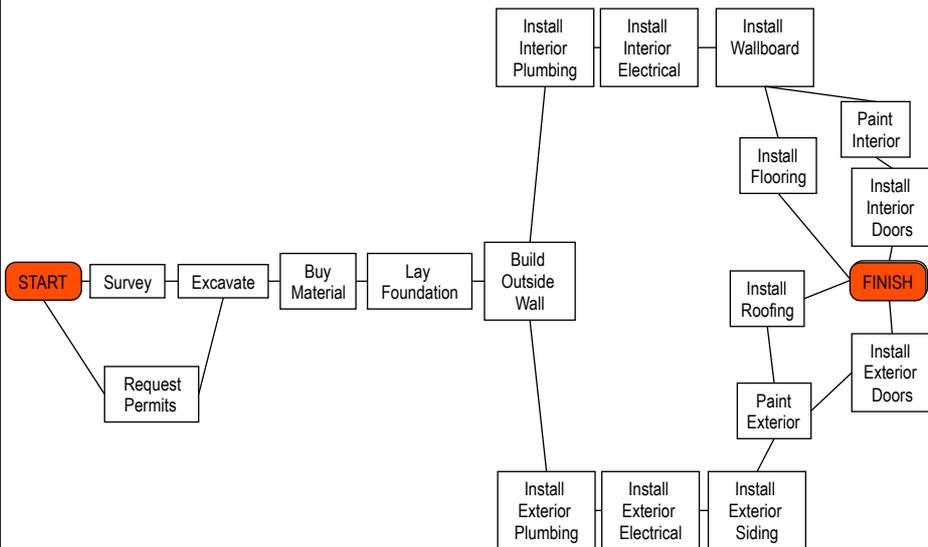
- A **task** describes the smallest amount of work tracked by management
- Typically 3-10 working days effort

- Tasks descriptions
 - Role
 - Work product
 - Start date
 - Planned duration
 - Required resources.

Example: Tasks for building a House



Example: Tasks for building a house



Tasks and Work Packages

- A task is specified by a **work package**
 - Description of work to be done
 - Preconditions for starting, duration, required resources
 - Work products to be produced, acceptance criteria for it
 - Risks involved
- A task must have **completion criteria**
 - Includes the acceptance criteria for the work products (deliverables) produced by the task.

Work Products

- A work product is a visible outcome of a task
- Examples
 - A document
 - A review of a document
 - A presentation
 - A piece of code
 - A test report
- Work products delivered to the customer are called **deliverables**

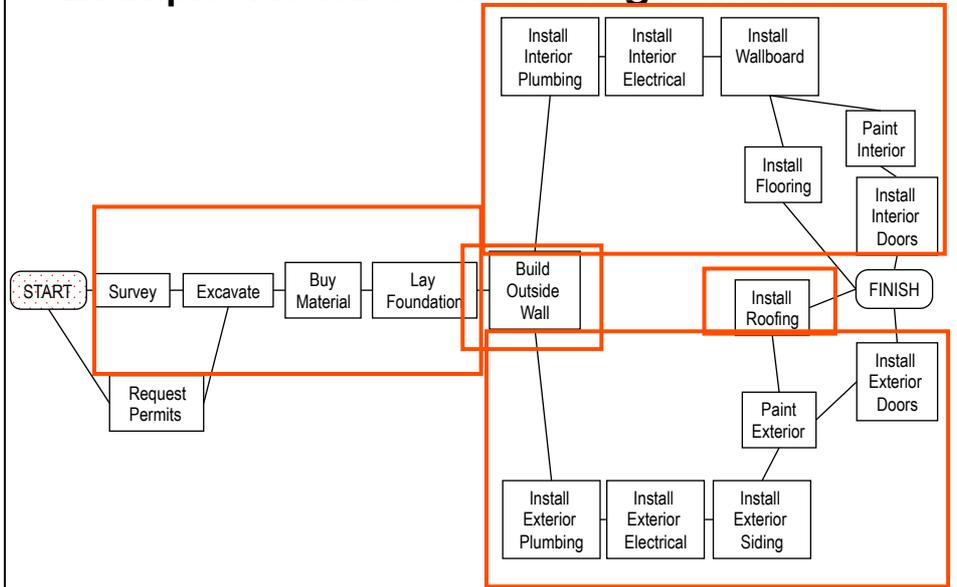
Task Sizes

- Tasks are decomposed into sizes that allow monitoring
 - You may not know how to decompose the problem into tasks at first
 - Depends on the nature of work and how well task is understood.
- Finding the appropriate size is crucial
 - To-do lists from previous projects
 - Each software development activity identifies more tasks and modifies existing ones.

Activities

- Major unit of work
- Culminates in a major project milestone:
 - Scheduled event used to measure progress
 - Internal checkpoints should not be externally visible
 - A project milestone usually produces a baseline
- Activities are often grouped again into higher-level activities with different names:
 - Phase 1, Phase 2 ...
 - Step 1, Step 2 ...
- Allows separation of concerns
- Precedence relations can exist among activities
 - Example: “A1 must be executed before A2”

Example: Activities for Building a House

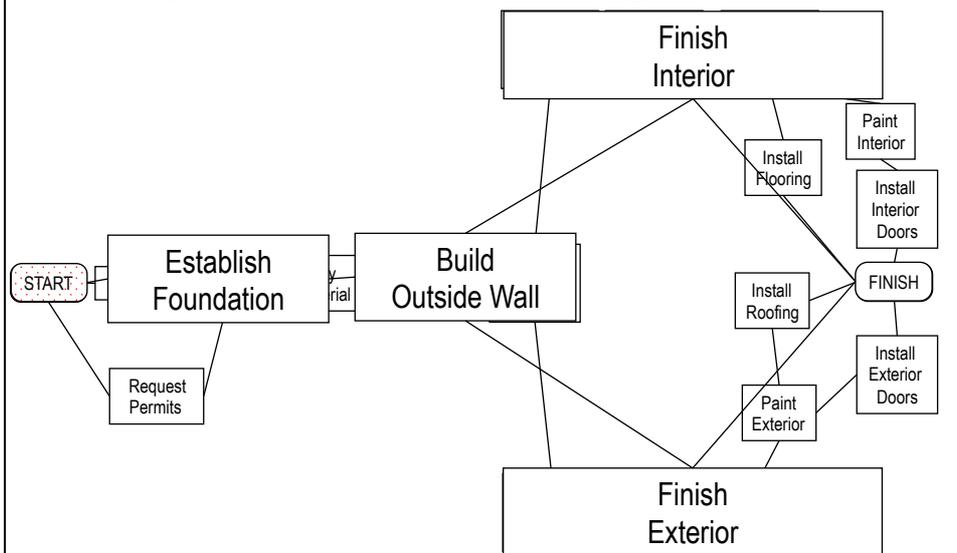


Bernd Bruegge & Allen H. Dutoit

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

27

Example: Activities for Building a House



Bernd Bruegge & Allen H. Dutoit

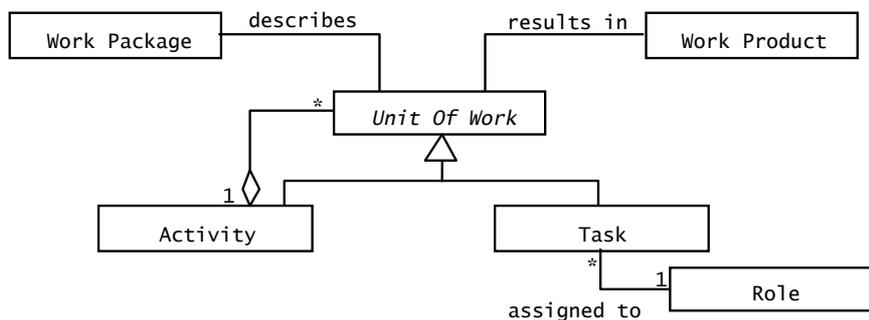
Object-Oriented Software Engineering: Using UML, Patterns, and Java

28

Examples of Software Engineering Activities

- Planning
- Requirements Elicitation
- Analysis
- System Design
- Object Design
- Implementation
- Testing
- Delivery

Associations between Tasks, Activities, Roles, Work Products, and Work Packages



Summary

- Projects are concerted efforts towards a goal that take place within a limited time
- Project participants are organized in terms of teams, roles, control relationships, and communication relationships.
- An individual can fill more than one role.
- Work is organized in terms of tasks assigned to roles and producing work products.