

Introduction to Software Engineering

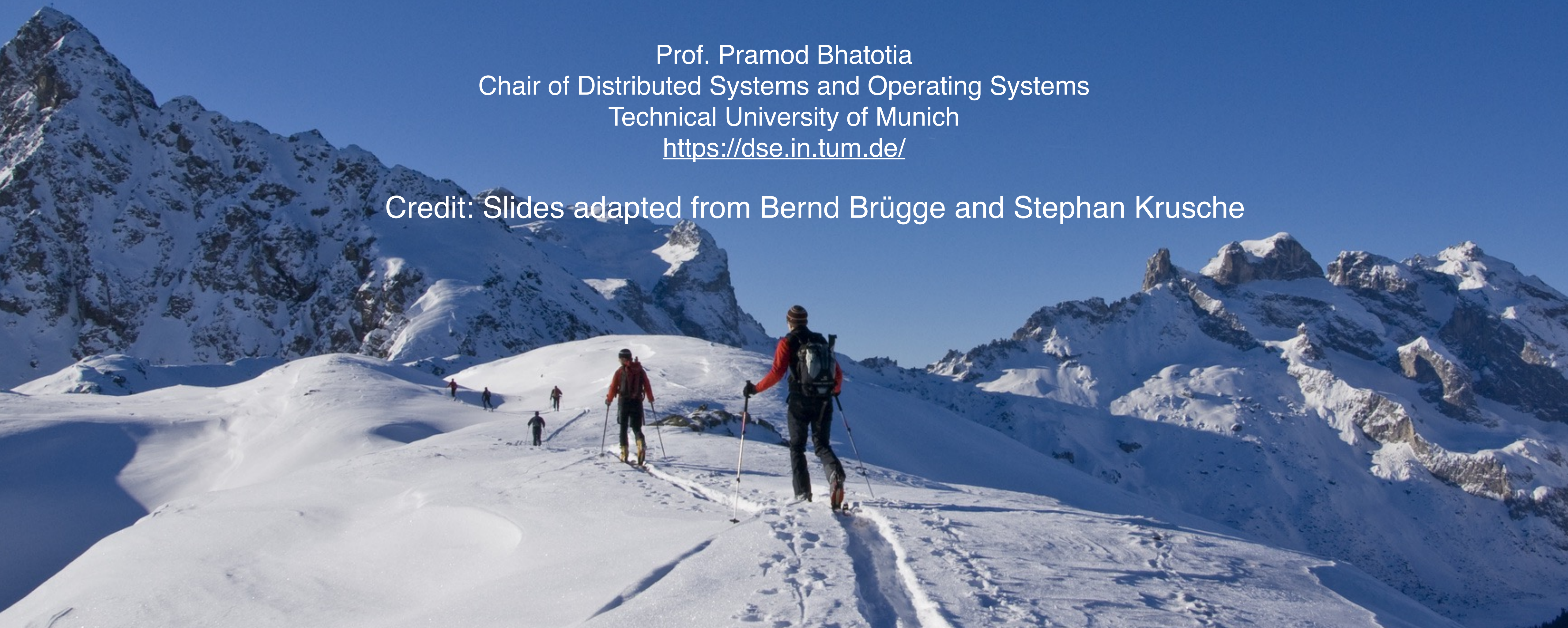
# 10b Project Management

Pramod Bhatotia



Prof. Pramod Bhatotia  
Chair of Distributed Systems and Operating Systems  
Technical University of Munich  
<https://dse.in.tum.de/>

Credit: Slides adapted from Bernd Brügge and Stephan Krusche





# Roadmap of the Lecture



- **Context and assumptions**
  - We completed all software development lifecycle activities
- **Learning goals: at the end of this lecture you are able to**
  - Project management
  - Work breakdown structure
  - Organization forms
  - Communication

# Outline

## **Project management**

- Work breakdown structure
- Organization forms
- Communication

# Definition: **project**

- Unique undertaking, limited in time, with a clear goal and a specific budget, requiring a concerted effort to create a product or service
- Consists of
  - A start date and duration
  - A set of **deliverables** for a client
  - A **schedule**
  - All technical and managerial **activities** required to produce and deliver the deliverables
  - **Resources** consumed by the activities
- Managed by a **project manager** who
  - Administers the resources
  - Maintains accountability
  - Makes sure the project goals are met

# Typical project management issues

- How should the project be **organized**?
- **Who** should be part of it?
- How do we **break down** the overall **work** to be done?
- How do we **schedule** the work?
- What are the **deliverables**?
- Who should do what? —> **Roles**

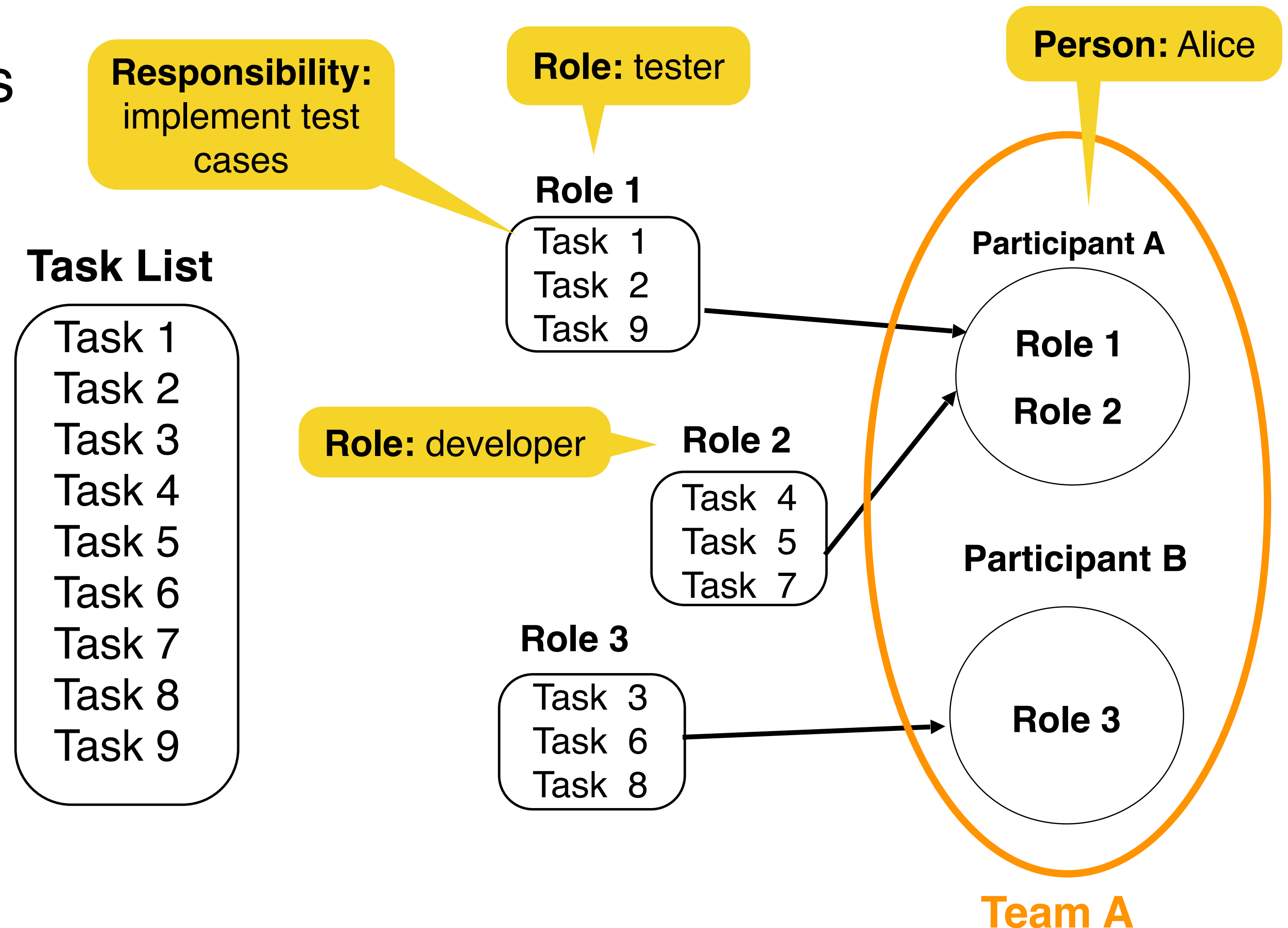
Defines a set of **responsibilities**: duties or tasks a person is assigned to do

[Examples](#) of roles and corresponding responsibilities

- **Project manager**
  - Administer the resources
  - Make sure the project goals are met
- **Analyst**
  - Analyse the application domain
  - Create a taxonomy of the domain abstractions
- **System architect**
  - Decompose the system into subsystems
  - Choose a software architectural style
- **Tester**: design and implement tests

# Roles and responsibilities

- Responsibilities (e.g. in the form of specific tasks) are assigned to roles
- Roles are assigned to people
- People are assigned to teams



# Assignment of roles to participants

- **One to one:** ideal but rare
- **Many to few**
  - Each project member assumes several "hats"
  - Danger of over-commitment
- **Many to “too many”**
  - Some people don't have significant roles
  - Lack of accountability
  - Losing touch with project
- **Problems** in role assignments
  - **Incompetence:** the wrong person fills the wrong role
  - **Useless role:** the role exists only to minimize damage control
  - **Increase of bureaucracy:** the role swells unnecessarily



# Key concepts for mapping roles to people



- **Authority:** the ability to make binding decisions between people and roles
- **Responsibility:** the commitment of a role to achieve specific results
- **Accountability:** tracking a performance of a task to a specific person
- **Delegation:** binding a task assigned to one person to another person

# Delegation in project management

- Binding a task assigned to one person to another person
  - 3 main reasons for delegation
    - 1) **Time management:** free yourself up for other tasks
    - 2) **Expertise:** the most qualified person makes the decision
    - 3) **Training:** develop another person's ability to handle additional assignments
- ➡ **You can delegate work, but you cannot delegate responsibility**
- ➡ **You can only share responsibility**

# Outline

- Project management

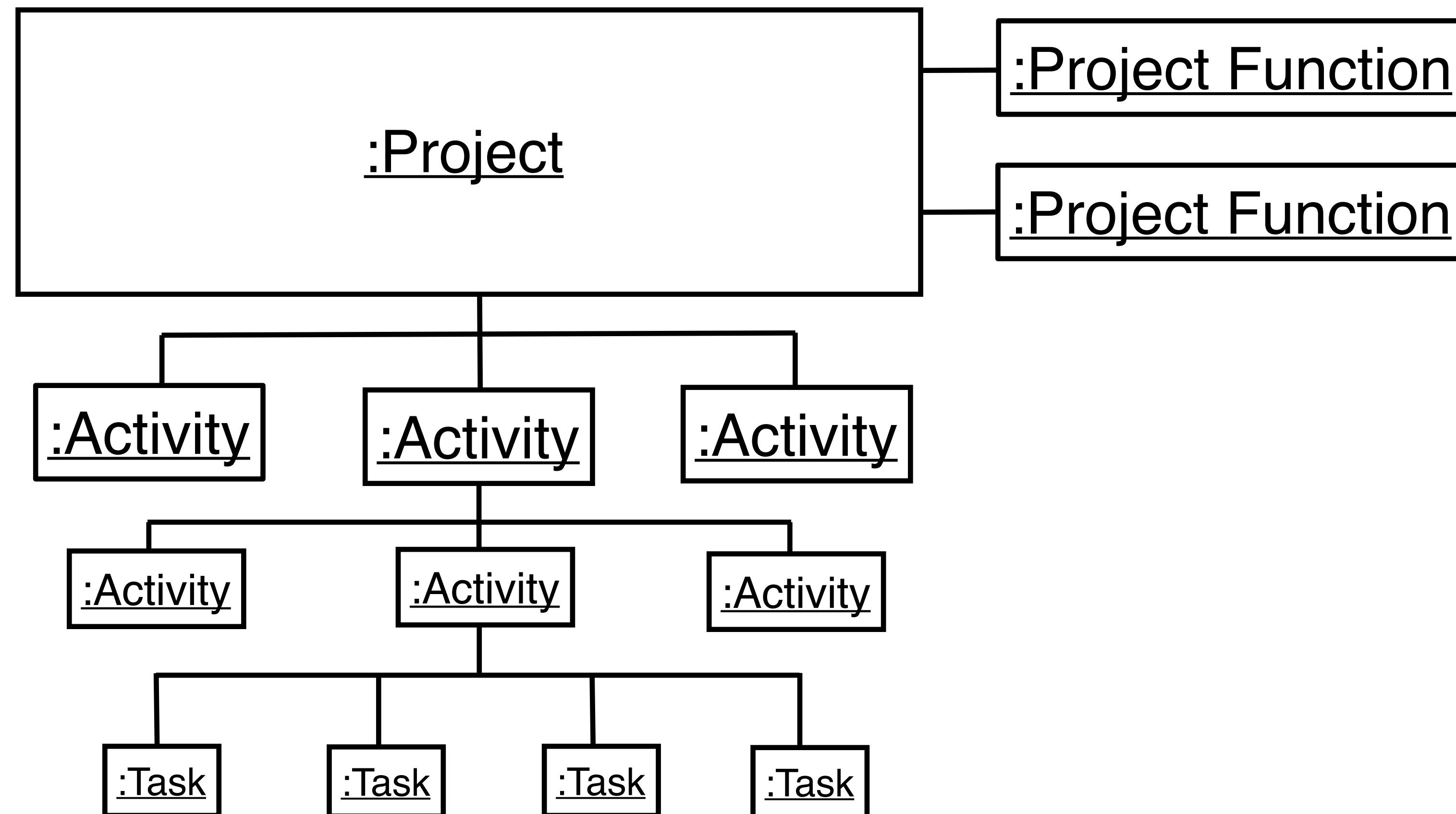
## **Work breakdown structure**

- Organization forms
- Communication



# Example of a project's work breakdown structure

A project includes project functions, activities and tasks



- Major work that culminates in a **project milestone**
  - A project milestone is a scheduled event used to visualize/measure progress
  - A project milestone is visible to the customer
  - A project milestone usually produces a baseline
- Can have internal checkpoints (not externally visible)
- There is often a precedence relation
  - **Example:** “activity A1 must be finished before activity A2 can start”

# Example activities in a software project



- Requirements elicitation
- Analysis
- System design
- Implementation
- Testing

Some of these activities span the duration of a project —> **project functions**

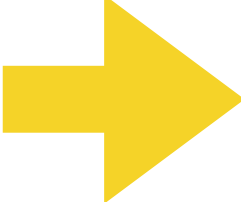


# Project function

- An activity that spans the entire duration of a software project
- **Examples** of project functions include
  - Project management
  - Software configuration management
  - Quality management
  - Continuous integration
  - Release management

- Describes the smallest amount of work monitored by the project manager
- Typically less than 2-4 working days effort
- Associated with
  - Role
  - Work package
  - Work product
  - Start date
  - Duration
  - Required resources

# Work package

- A **task** or **activity** is specified by a **work package** which contains
  - The description of work to be done
  - Preconditions for starting, duration, required resources
  -  • **Work products** to be produced
  - Acceptance criteria
  - Involved risks
- A work package must have completion criteria
  - Includes the acceptance criteria for the work products produced by the task or activity



# Work product



- A tangible outcome of work
- Examples
  - A model
  - A review of a document
  - A presentation
  - A piece of code
  - A test report
- Work products that are given to the customer are called deliverables

# Outline

- Project management
- Work breakdown structure

## **Organization forms**

- Communication

# Project organization



- Defines the relationships among resources (in particular participants) in a project
- A project organization should define
  - Who decides what (**decision structure**)
  - Who reports their status to whom (**reporting structure**)
  - Who communicates with whom (**communication structure**)
- 3 types
  - Functional organization
  - Project-based organization
  - Matrix organization

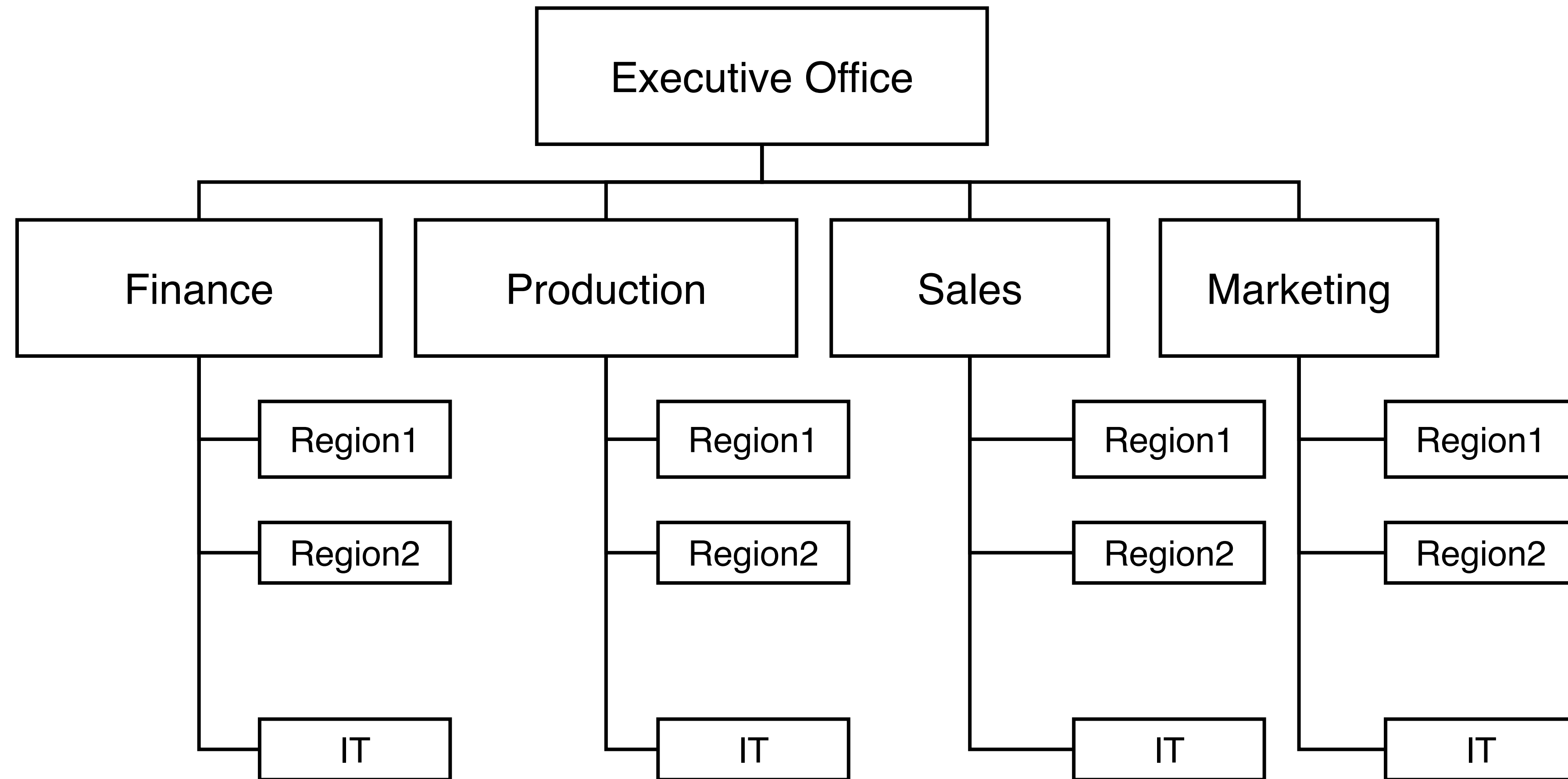


# Functional organization



- People are grouped into departments, each of which addresses one activity (“function”)
- Examples of departments
  - In traditional companies: finance, production, sales, marketing
  - In software companies additionally: analysis, design, integration, testing, delivery

# Example of a functional organization



Also called **line organization**

# Properties of functional organizations



- **Advantage**
  - + Members of a department have a good understanding of the functional area they support
- **Disadvantages**
  - High chance of work duplication or overlap among departments

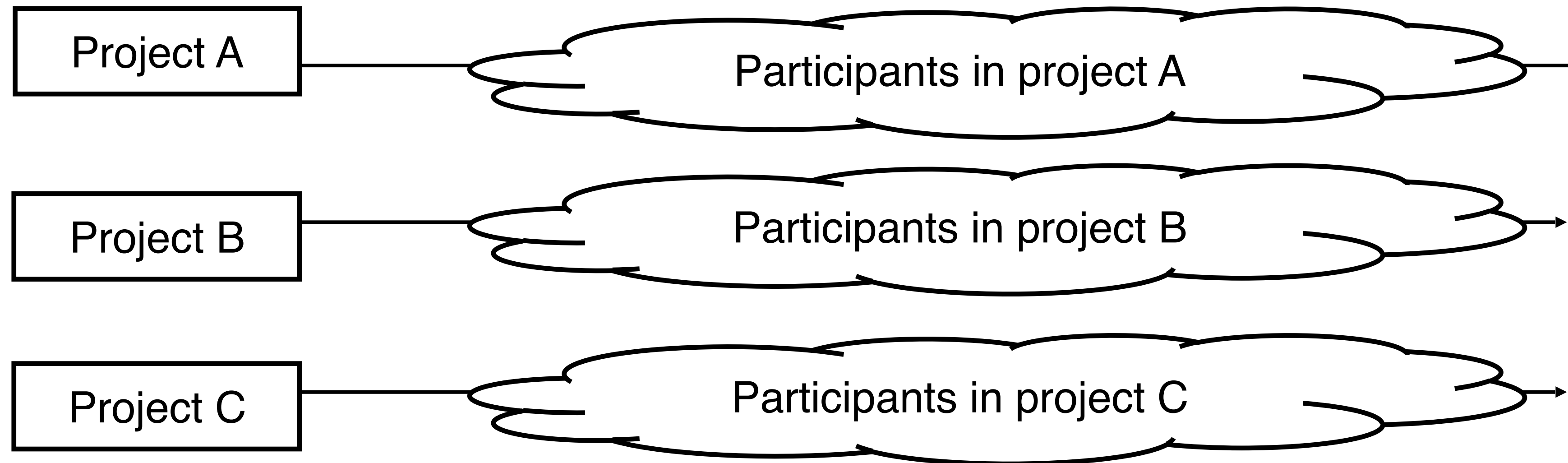


# Project-based organization



- People are assigned to **one of the several project** in the organization, each of which has a problem to be solved in a certain time within a given budget
- Key properties of project-based organizations
  - Teams are assembled when a project is created
  - Each project has a project manager
  - A participant is involved only in a single project
  - Teams are disassembled when the project terminates

# Example of project-based organization



# Properties of project-based organizations



- **Advantages**

- + Responsive to new requirements  
(the project is newly established and can be tailored around the problem)
- + New people familiar with the problem or with special capabilities can be hired
- + There is no idle time for the project members

- **Disadvantages**

- Teams cannot be assembled rapidly: often difficult to manage the staffing/hiring process (flat staffing vs. gradual staffing)
- Roles and responsibilities need to be defined at the beginning of each project (because there are no predefined departments as in a functional organization)

# When to use which organization type?

- **Functional organization**

- Projects with high degree of certainty, stability, uniformity and repetition
- Requires little communication
- Role definitions are clear
- The more people on a project, the more the need for a formal structure

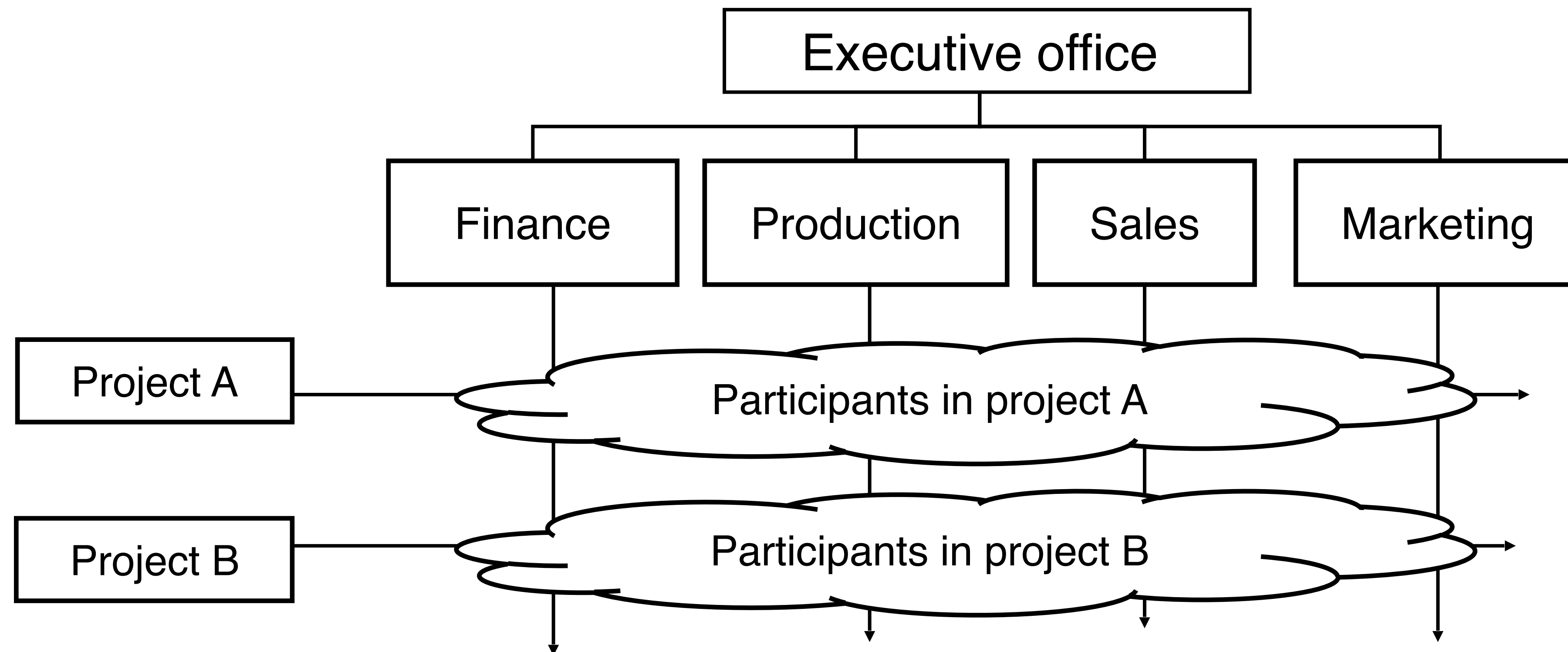
- **Project-based organization**

- Project has high degree of uncertainty
- Open communication is needed among participants
- Roles are defined on project basis
- Requirements are likely to change during the project
- A new technology that could affect the outcome may appear during the project



# Matrix organization

- People from different departments of a functional organization are assigned to work on one or more projects



# Properties of matrix organizations



- **Advantages**

- + Teams for projects can be assembled rapidly from the departments
- + Expertise can be applied to different projects as needed
- + Consistent reporting and decision procedures can be used for projects of the same type

- **Disadvantages**

- Team members are often not familiar with each other
- Team members have different working styles

# Challenges in matrix organizations

- Team members working on multiple projects have competing demands for their time
- Multiple work procedures and reporting systems are used by different team members
- **Double boss problem:** team members must respond to two different bosses with different focuses
  - **Focus of the department manager:** assignments to different projects, performance appraisal
  - **Focus of the project manager:** work assignments to project members, support of the project team, deliver project in time and within budget
- Department and project interests might be in conflict with each other

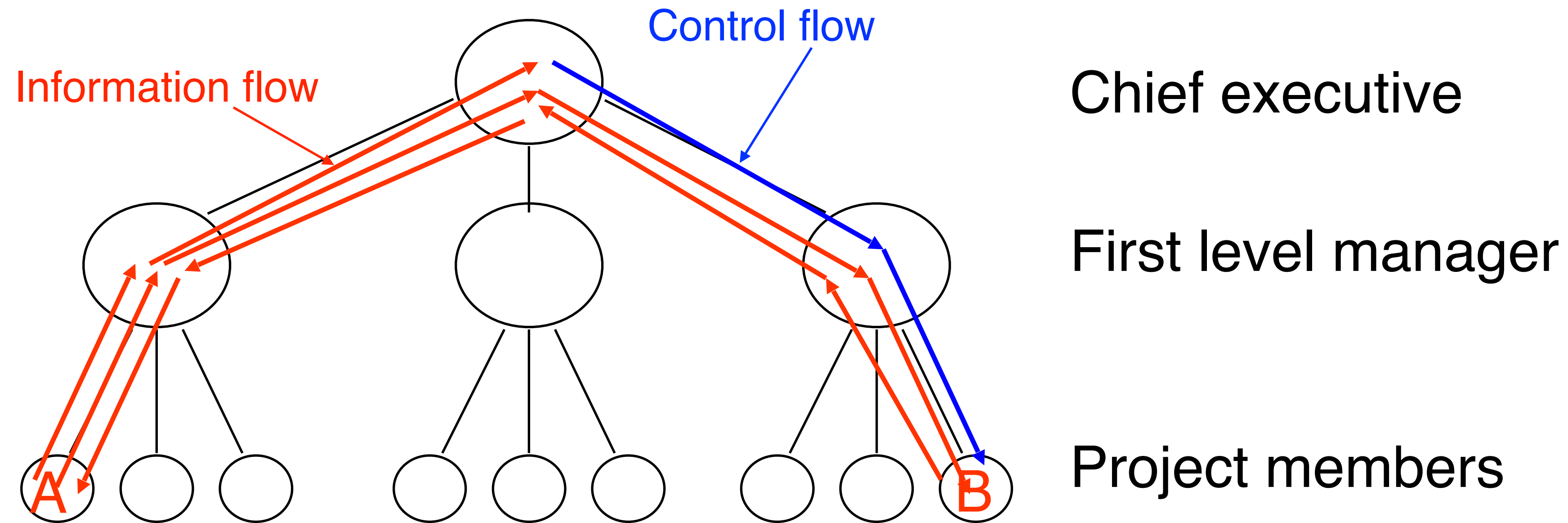
# Project organization structures



- A project organization has at least 3 structures that model the relationships between people
  - 1) **Decision structure** models the control flow: who decides what?
  - 2) **Reporting structure:** who reports their status to whom?
  - 3) **Communication structure** models the information flow: who facilitates communication with whom?



# Example of information and control flow in a line organization



A wants to talk to B: complicated communication flow

A wants to make sure B makes a certain change: complicated decision flow

Information and control flow along hierarchical boundaries

- Information flow in a hierarchical project organization does not work well with unexpected changes
- The manager is not necessarily always right and might even misunderstand communication requests
- Improving information flow through **non-hierarchical project organizations**
  - + Cut down bureaucracy (direct communication is possible)
  - + Reduce development time
  - + Better communication between multiple teams
  - + Decisions are expected to be made at each level
  - Hard to manage (who is in control in case of conflicts?)

# Outline

- Project management
- Work breakdown structure
- Organization forms

 **Communication**

- Project managers **and** software engineers need to acquire several skills
    - **Collaboration:** negotiate requirements with the client and with members from your team and other teams
    - **Presentation:** present a major part of the system during a review
    - **Technical writing:** write a part of the proposal, or a part of the project documentation
    - **Management:** facilitate a team meeting, find compromises, negotiate between conflicting demands
- In large system development efforts, you will spend more time communicating than coding
- **Technology manager**

- Clear and accurate communication is critical for the success of a project
- Modes of communication (also called communication events)
  - Planned communication
  - Event-driven communication
- Difference between communication events and communication mechanisms



# Communication event vs. mechanism

**Communication event:** information exchange with defined objectives & scope

- **Scheduled:** planned communication  
**Examples:** review, meeting
- **Unscheduled:** event-driven communication  
**Examples:** request for change, clarification, bug report

**Communication mechanism:** tool or procedure that can be used to deal with a communication event

- **Synchronous:** same time
- **Asynchronous:** different time

Another distinction can be made: **formal** vs. **informal** communication

# Communication events (**examples**)

- **Problem definition:** focus on scope
  - **Objective:** present goals, requirements and constraints
  - **Example:** client presentation
  - Usually scheduled at the beginning of a project
- **Project review:** focus on system models
  - **Objective:** assess status and review the system model
  - **Example:** analysis review, system design review
  - Scheduled after each project milestone
- **Client review:** focus on requirements
  - **Objective:** brief the client, agree on requirements changes
  - **Example:** requirements review, prototype review
  - The first client review is usually scheduled after the analysis phase

# Communication mechanisms (synchronous **examples**)



- **Informal meeting**

- **Example:** meeting at the coffee machine, hallway meeting
- **Supports:** unplanned conversations, request for clarification, request for change
- + Cheap and effective for resolving simple problems
- Information loss, misunderstandings are frequent

- **Formal meeting**

- **Example:** face to face, telephone conference tool, video conference tool
- **Supports:** planned conversations, client review, project review, status review, brainstorming, issue resolution
- + Effective for issue resolutions and consensus building
- High cost (people, resources)

# Communication mechanisms (asynchronous **examples**)



- **E-Mail**

- **Supports:** release, change request, brainstorming
- + Ideal for planned and formal communication and announcements
- E-mail taken out of context can be misunderstood, sent to the wrong person or lost

- **Chats**

- **Supports:** release, change request, brainstorming
- + Suited for discussion among people who share a common interest; cheap (shareware available)
- Rather informal

- **Wikis**

- **Supports:** release, change request, inspections
- + Documents contain links to other documents
- Does not easily support rapidly evolving documents

# Summary



- Projects are concerted efforts towards a goal within a limited time
- Project participants are organized in terms of teams, roles, control and communication relationships
- An individual can fill more than one role
- Work is organized in terms of activities and tasks
  - Tasks are assigned to roles
  - Tasks produce work products
- 3 project organization forms: functional, project-based, matrix
- Communication is critical: formal vs. informal, mechanisms vs. events



- F. P. Brooks: The Mythical Man Month: Anniversary Edition: Essays on Software Engineering. Addison-Wesley, Reading, MA, 1995
  - Also available as PDF File: [https://www.researchgate.net/publication/220689892\\_The\\_Mythical\\_Man-Month\\_Essays\\_on\\_Software\\_Engineering](https://www.researchgate.net/publication/220689892_The_Mythical_Man-Month_Essays_on_Software_Engineering)
- G. M. Weinberg: The Psychology of Computer Programming, Van Nostrand, New York, 1971
- D. J. Paulish: Architecture-centric Software Project Management, SEI Series in Software Engineering, Addison-Wesley, 2001
- E. Raymond: The Cathedral and the bazaar, 1998 <http://manybooks.net/titles/raymondericother05cathedralandbazaar.html>