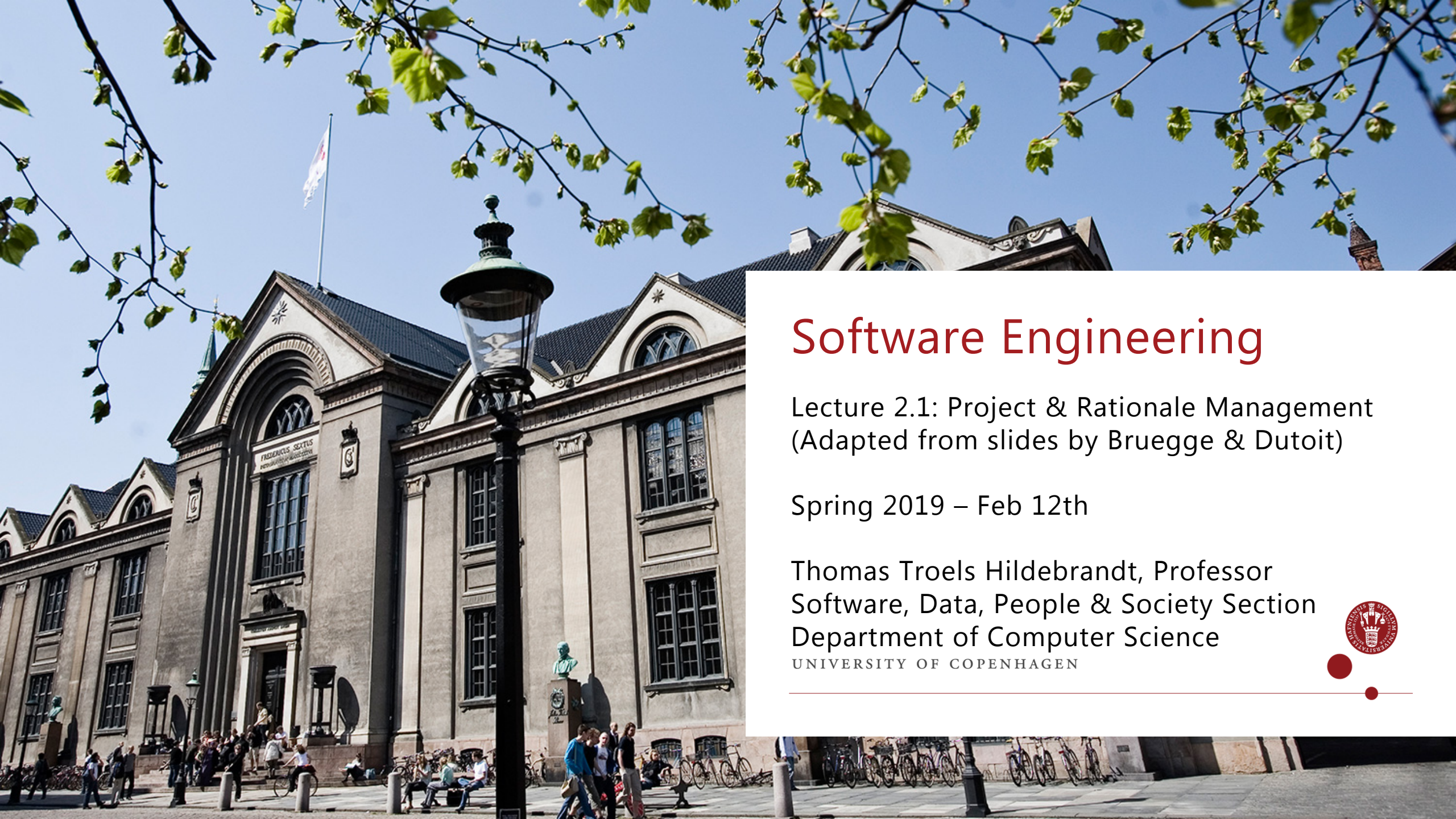


Software Engineering

Lecture 2.1: Project & Rationale Management
(Adapted from slides by Bruegge & Dutoit)

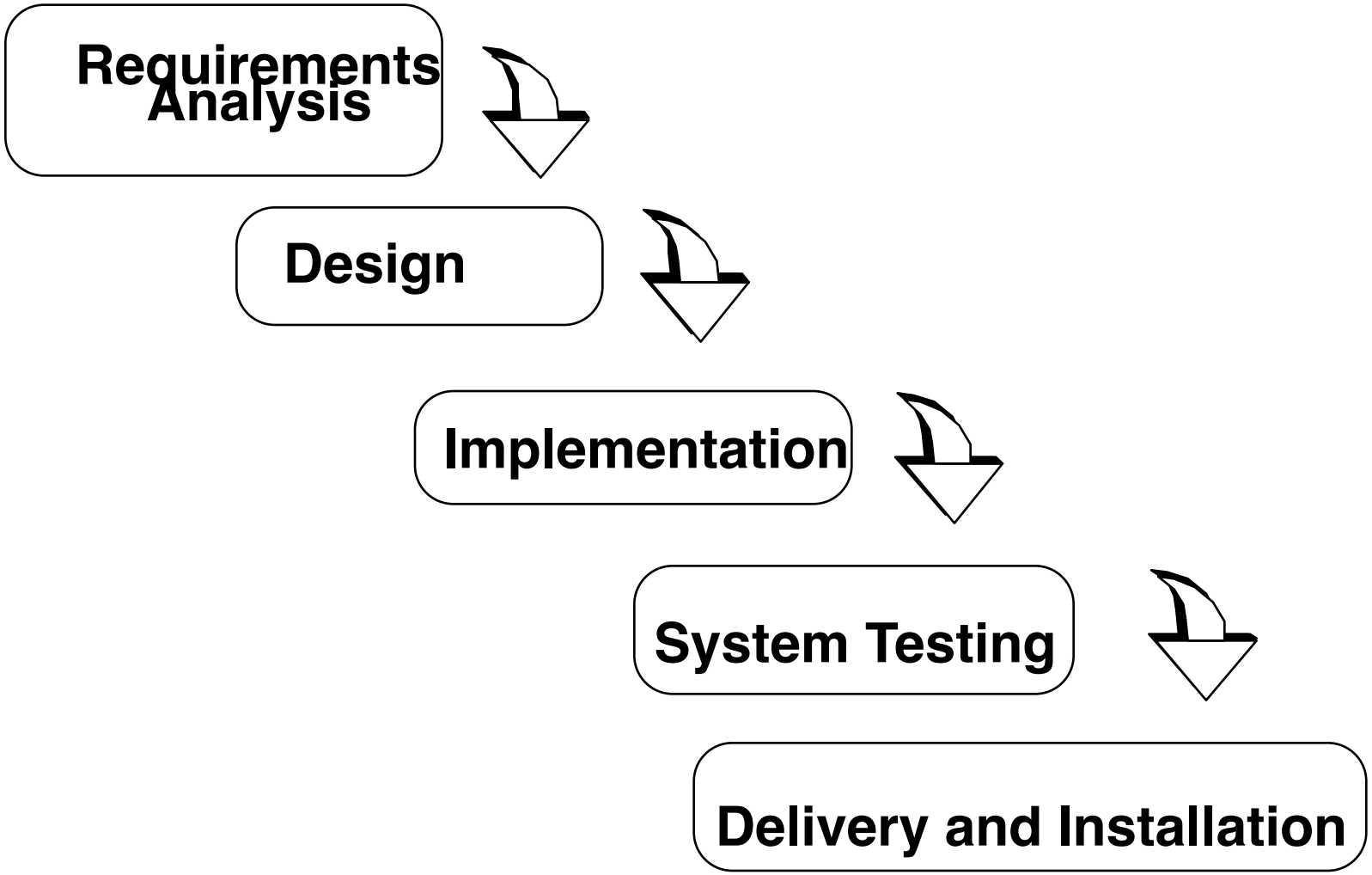
Spring 2019 – Feb 12th

Thomas Troels Hildebrandt, Professor
Software, Data, People & Society Section
Department of Computer Science
UNIVERSITY OF COPENHAGEN

Roadmap for today

- 9:15-9:40 Lecture 2.1: Project Organisation and Communication (Ch 3)
- 9:40-10 Exercises: Write first problem statement
- 10:15-10:40 Lecture 2.2: Rationale Magement (ch 12)
- 10:40-11 Exercises:
- 11:15-11.40 Lecture 1.3: Project Management (ch 14)
- 11:40-12.00 Exercises:

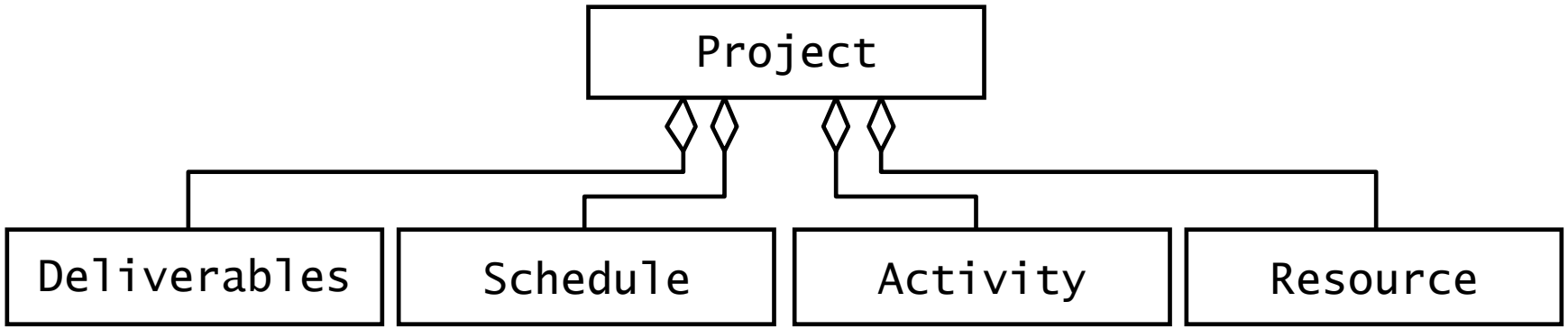
If Software Engineering was predictable



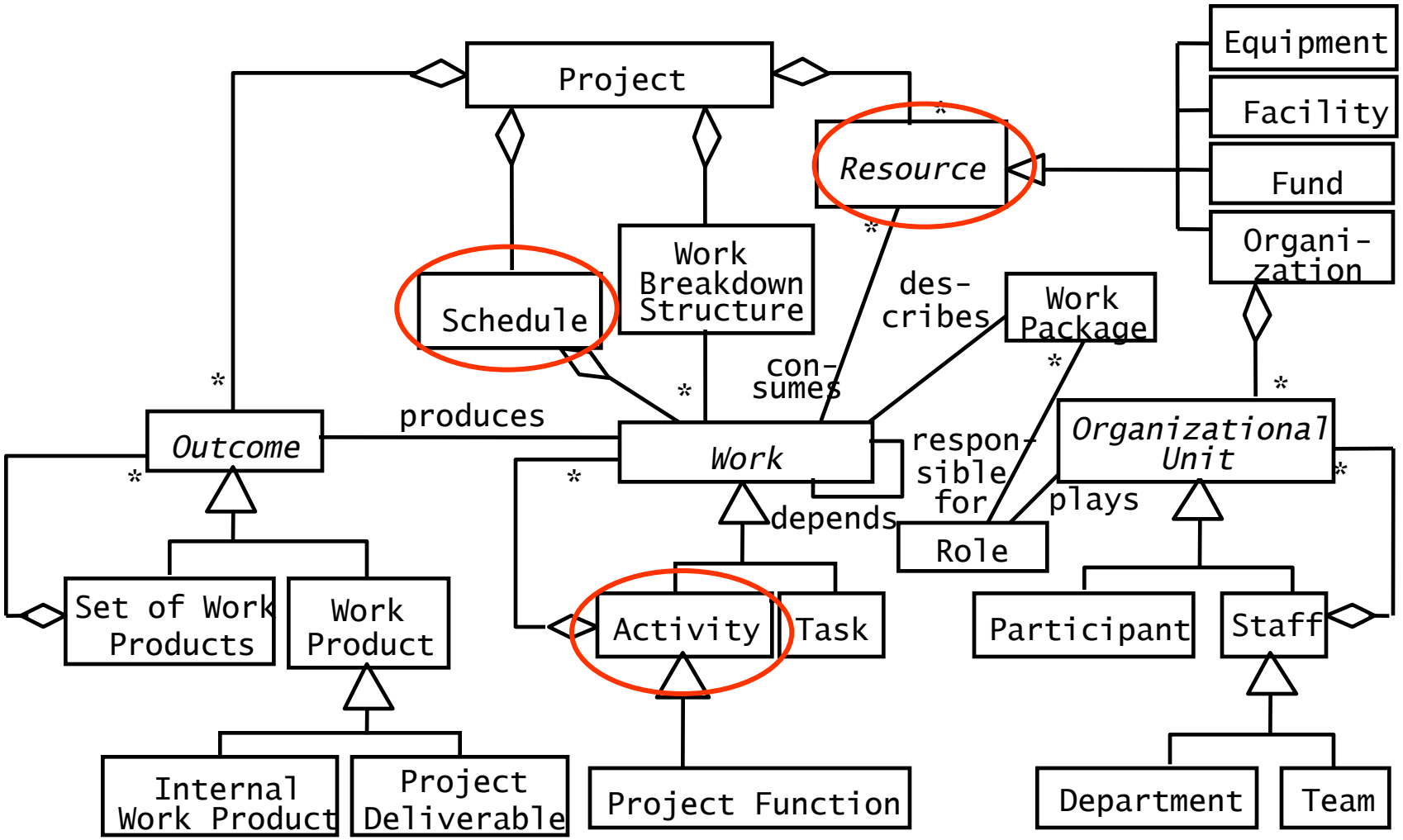
Collaborative Project Definition

- A (collaborative) project is an undertaking, limited in time, to achieve a set of goals that requires 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.

An UML model of a project

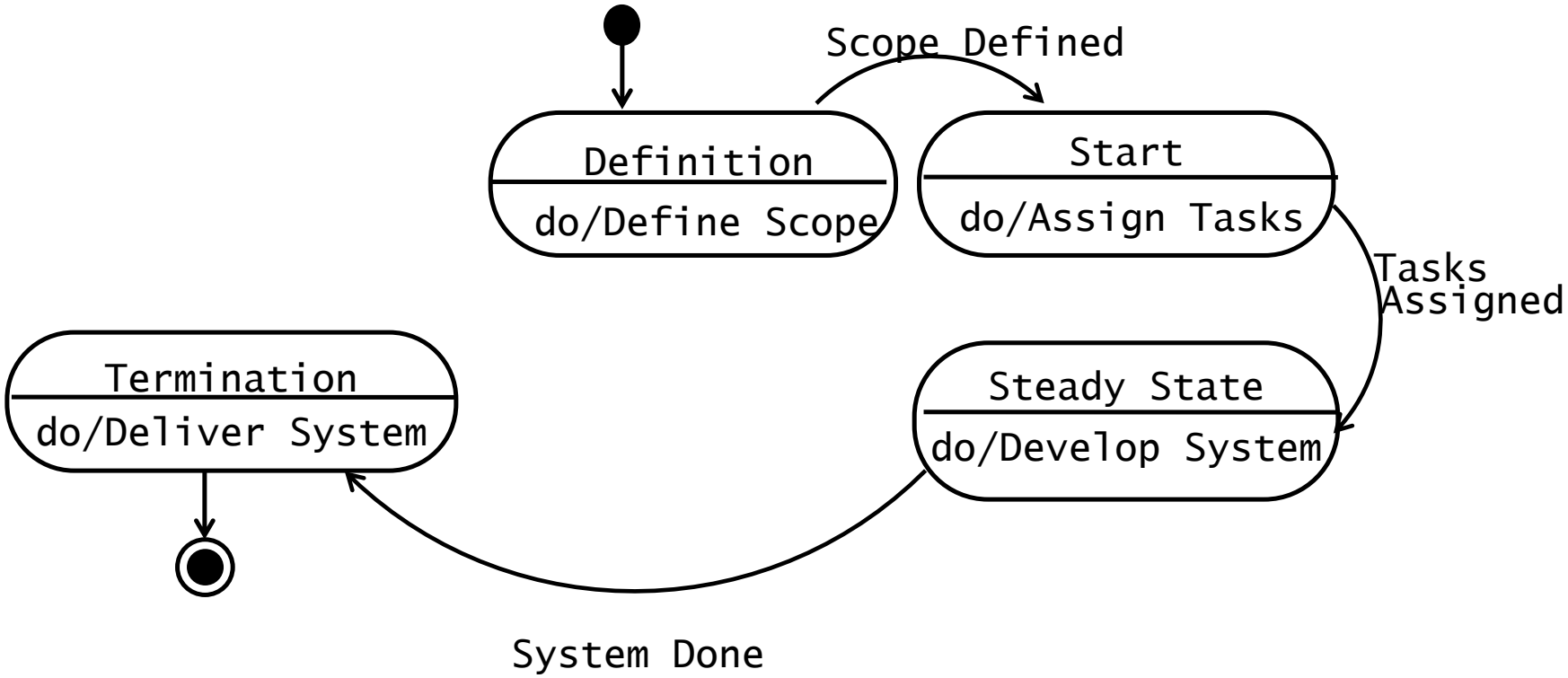


Project Model Refined



Did it help ?

Naive Dynamic Model of a project



Did it help ?

Project Organisation

- A project organization defines the relationships among resources, in particular the participants, in a project

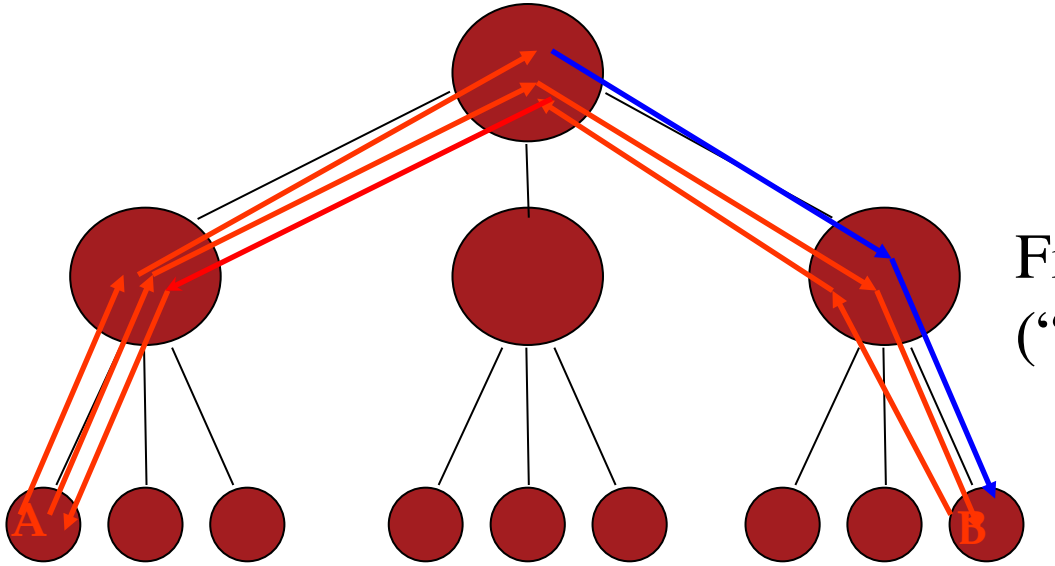
- A project organization should define
  - Who decides (decision structure)
  - Who reports their status to whom (reporting structure)
  - Who communicates with whom (communication structure)

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**

Hierarchical Project Organization and Communication



Chief Executive

First Level Manager
("Front-Line Manager")

Project Members

Pros ?

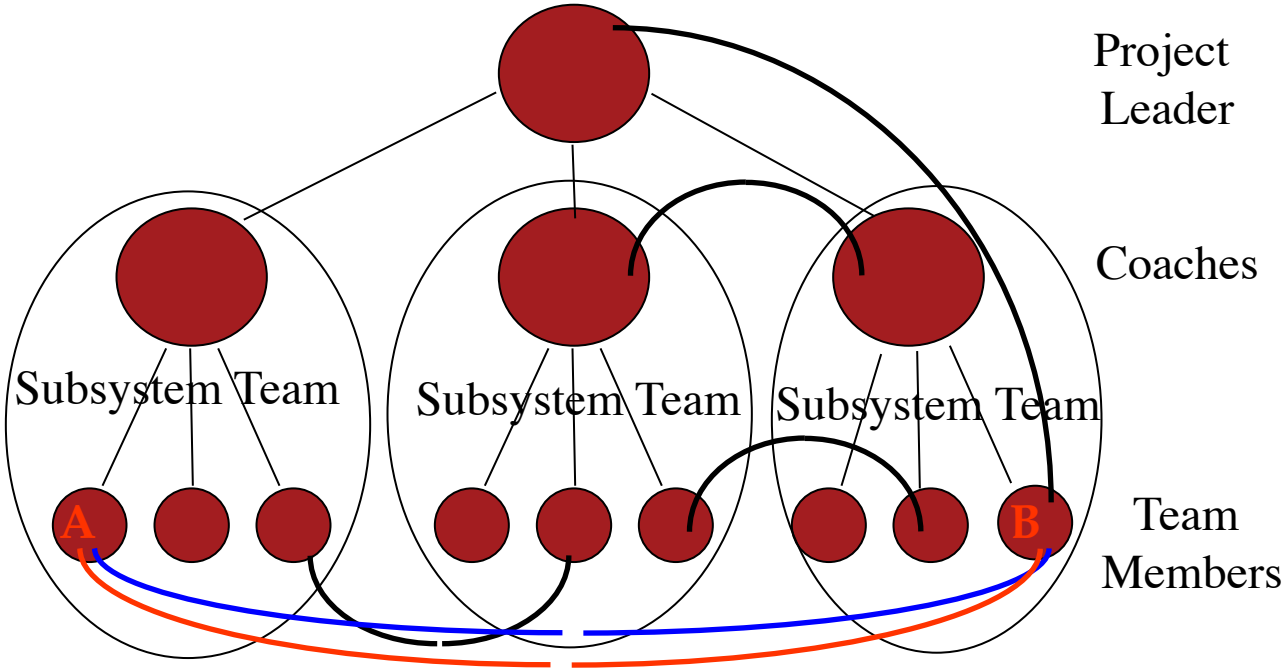
Cons ?

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

Peer-to-peer Communication



Pros ?

Cons ?

A wants to talk to B: Simple Information Flow

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

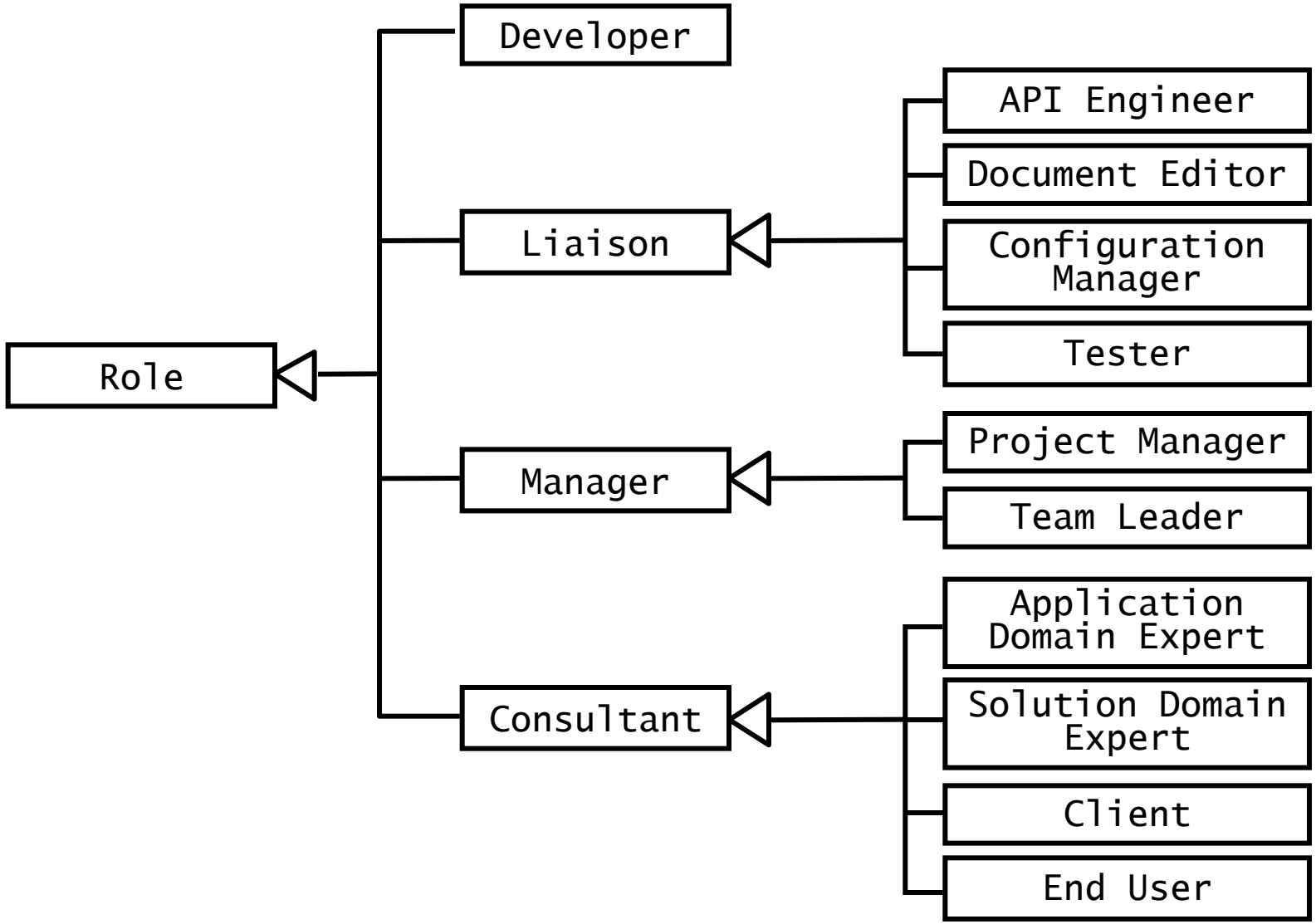
Communication is critical

- In large system development efforts, you will spend more time communicating than coding
- A software engineer needs to learn the so-called soft 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
  - Management
    - Facilitate a team meeting
  - Technical writing
    - Write part of the project documentation.

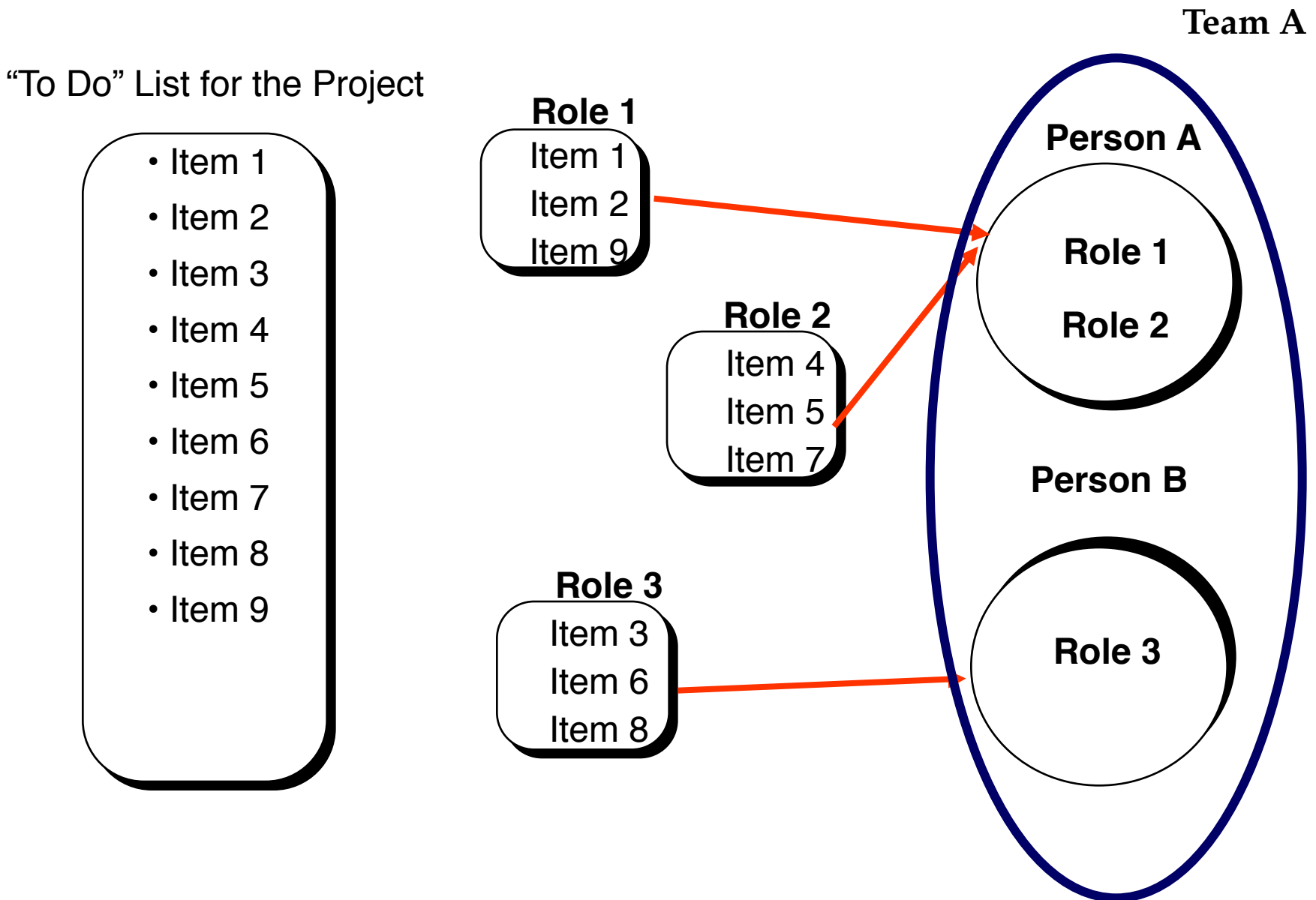
Roles

- 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 Development



Responsibilities for items/activities, Roles, People

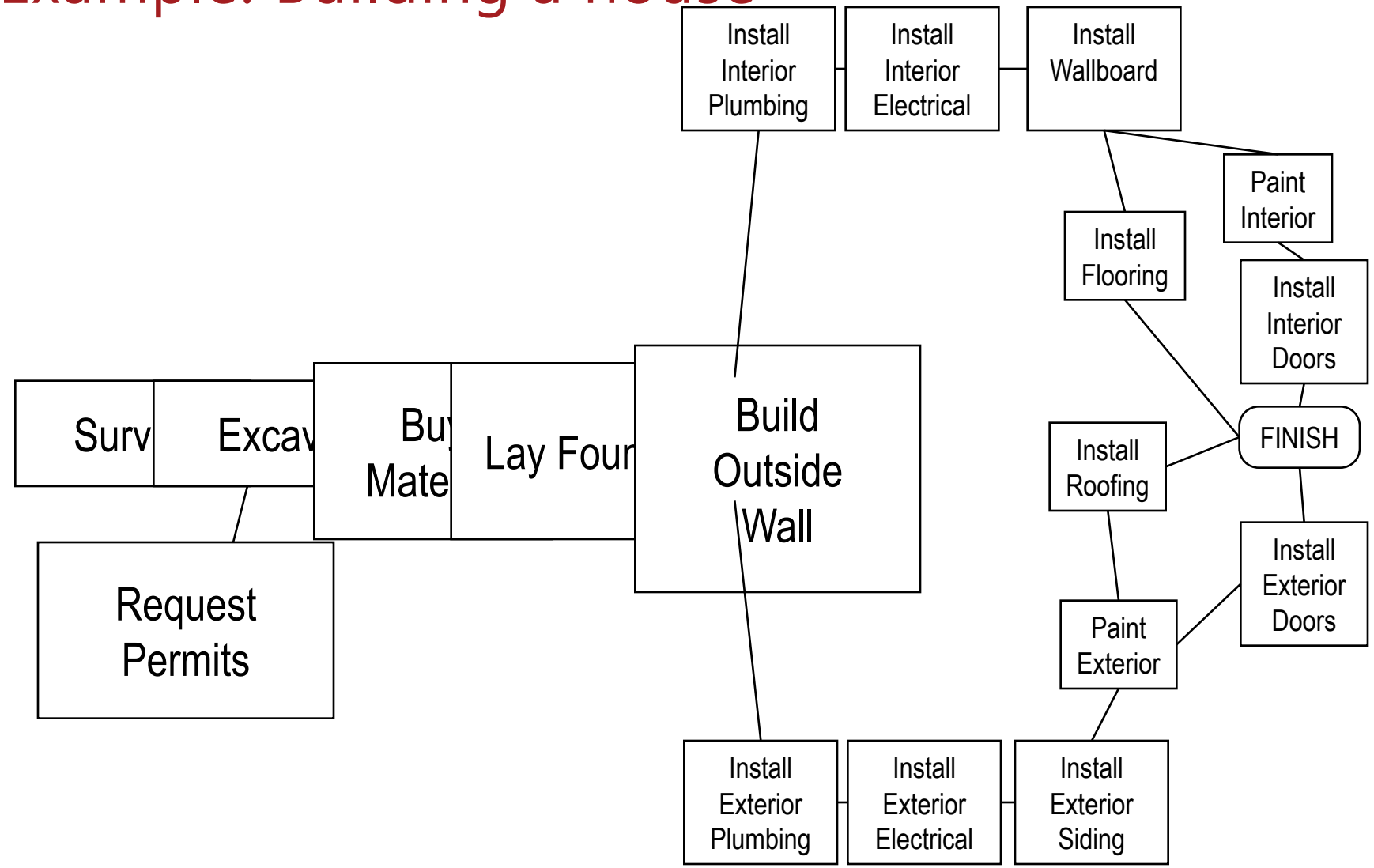


Tasks

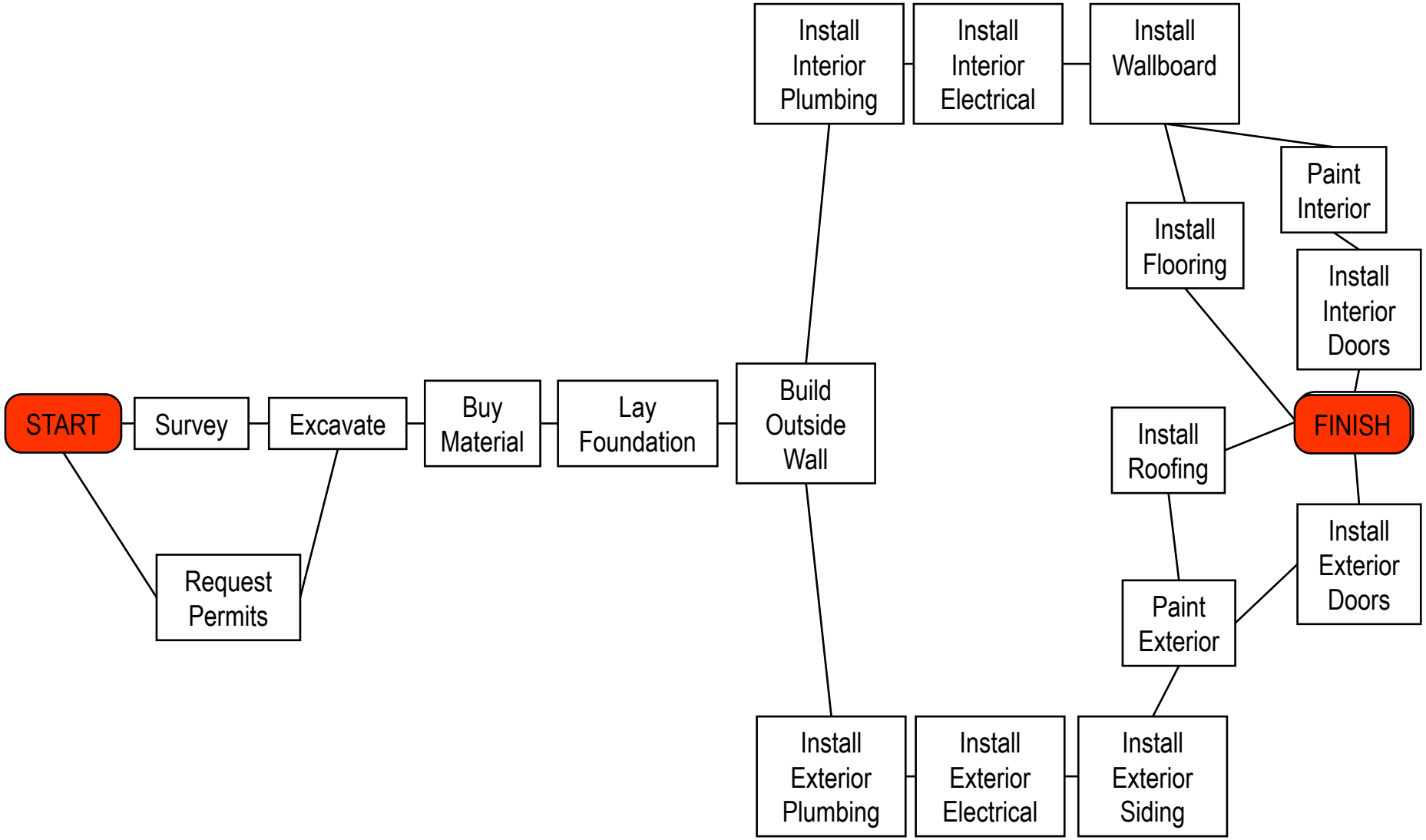
- A task describes the smallest amount of work tracked by management
- Typically 3-10 working days effort

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

Example: Building a house



Example: Ordering of tasks



Tasks and Work packages (hierarchy in the model)

- 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**

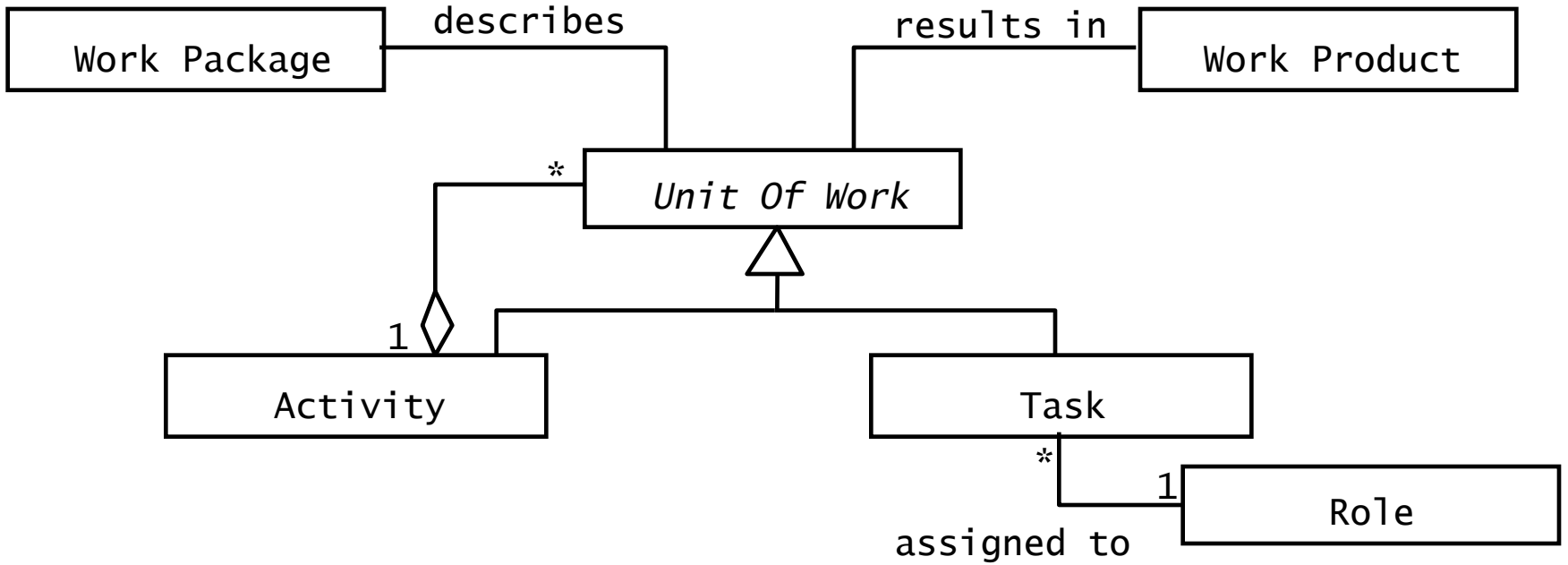
Task sizes

- 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.

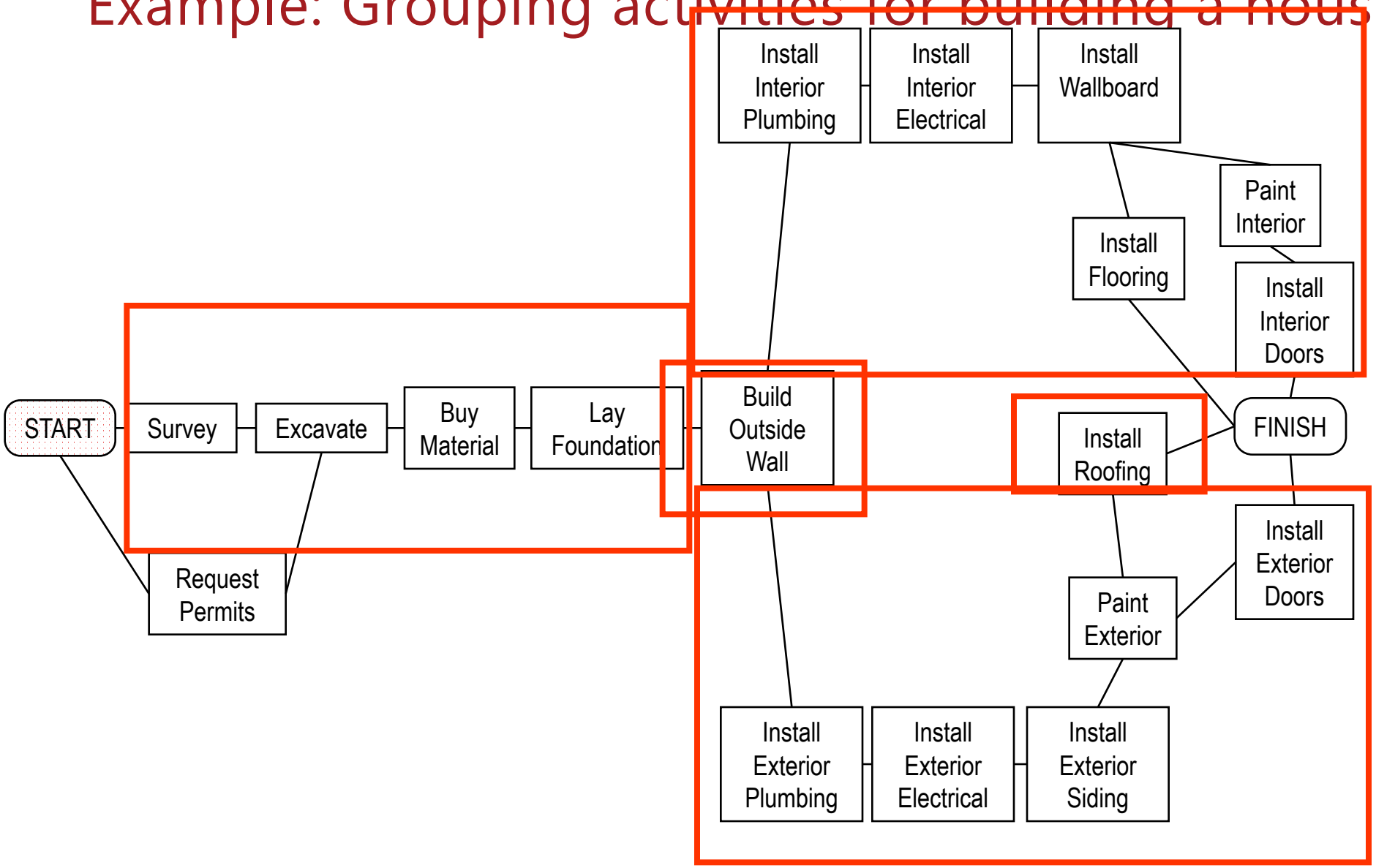
Activities

- 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 (PERT chart, DCR Graph)
  - Example: “A1 must be executed before A2”

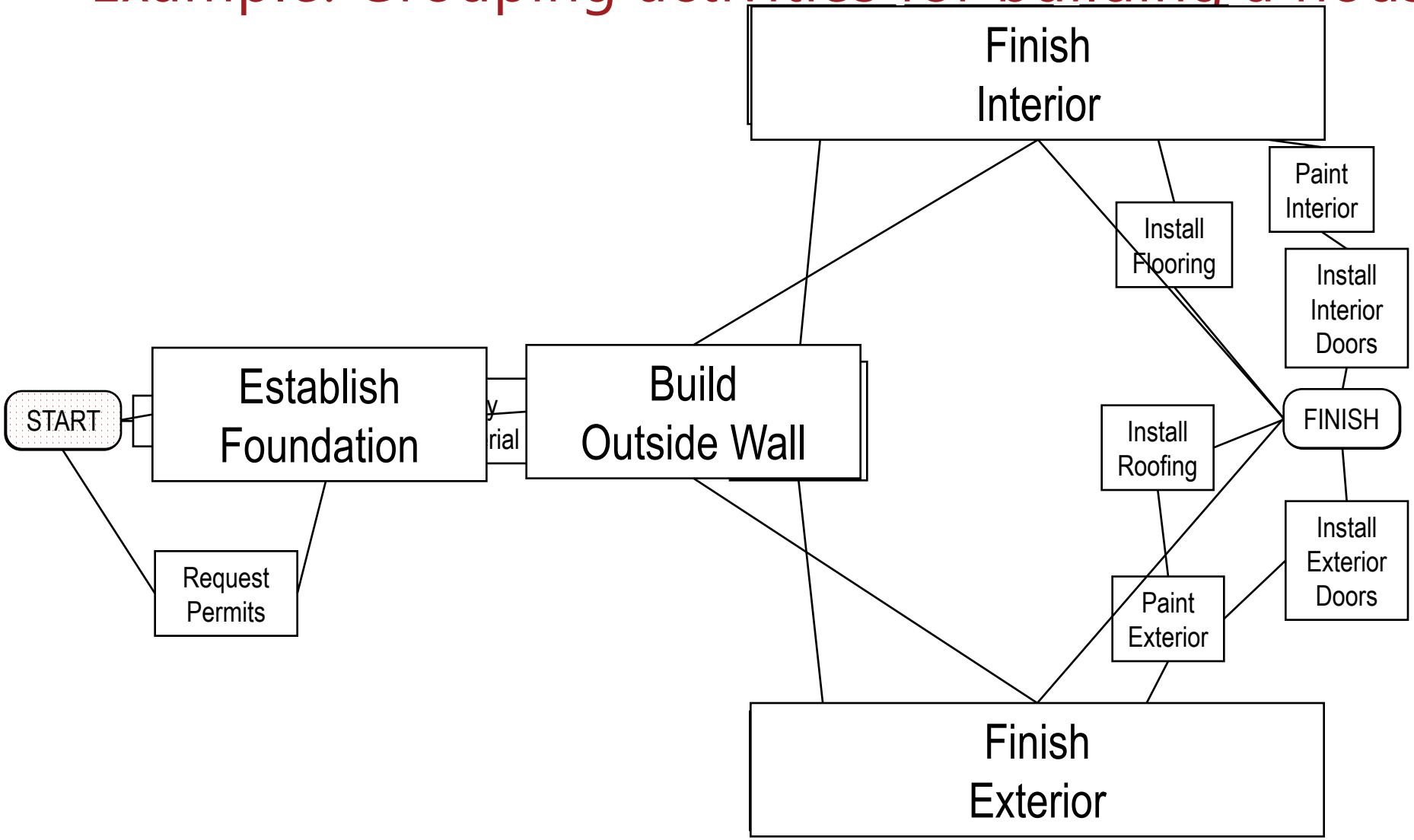
Associations between Tasks, Activities, Roles, Work Products, and Work Packages



Example: Grouping activities for building a house



Example: Grouping activities for building a house



Software Engineering Activities Examples

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

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.

Planned Software Engineering Communication Activities/Events

Problem Definition

- 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
- Examples: Analysis review, system design review
- Scheduled around project milestones and deliverables

Client Review: Focus on requirements

- Objective: Brief the client, agree on requirements changes
- The first client review is usually scheduled after analysis phase.

Planned Software Engineering Communication Activities/Events

Walkthrough (Informal)

- Objective: Increase quality of subsystem
- Example
  - Developer informally presents subsystem to team members (“peer-to-peer”)
- Scheduled by each team

Inspection (Formal)

- Objective: Compliance with requirements
- Example
  - Demonstration of final system to customer (Client acceptance test)
- Scheduled by project management

Planned Software Engineering Communication Activities/Events

Status Review

- Objective: Find deviations from schedule and correct them or identify new issues
- Example
  - Status section in regular weekly team meeting

Brainstorming

- Objective: Generate and evaluate large number of solutions for a problem
- Example
  - Discussion section in regular weekly team meeting.

Planned Software Engineering Communication Activities/Events

Release

- Objective: Baseline the result of each software development activity
- Examples:
  - Software Project Management Plan
  - Requirements Analysis Document
  - System Design Document
  - Beta version of software
  - Final version of software
  - User Manual
- Usually scheduled after corresponding activity (“phase”)

Postmortem Review

- Objective: Describe Lessons Learned
- Scheduled at the end of the project

Unplanned Communication Activities/Events

Request for clarification

- The bulk of communication among developers, clients and users
- Example: A developer may request a clarification about an ambiguous sentence in the problem statement.

```
From: Alice
Newsgroups: vso.discuss
Subject: SDD
Date: Wed, 2 Nov 9:32:48 -0400

When exactly would you like the System Design Document? There
is some confusion over the actual deadline: the schedule
claims it to be October 22, while the template says we have
until November 7.
Thanks,-Alice
```

Unplanned Communication Activities/Events

Request for change

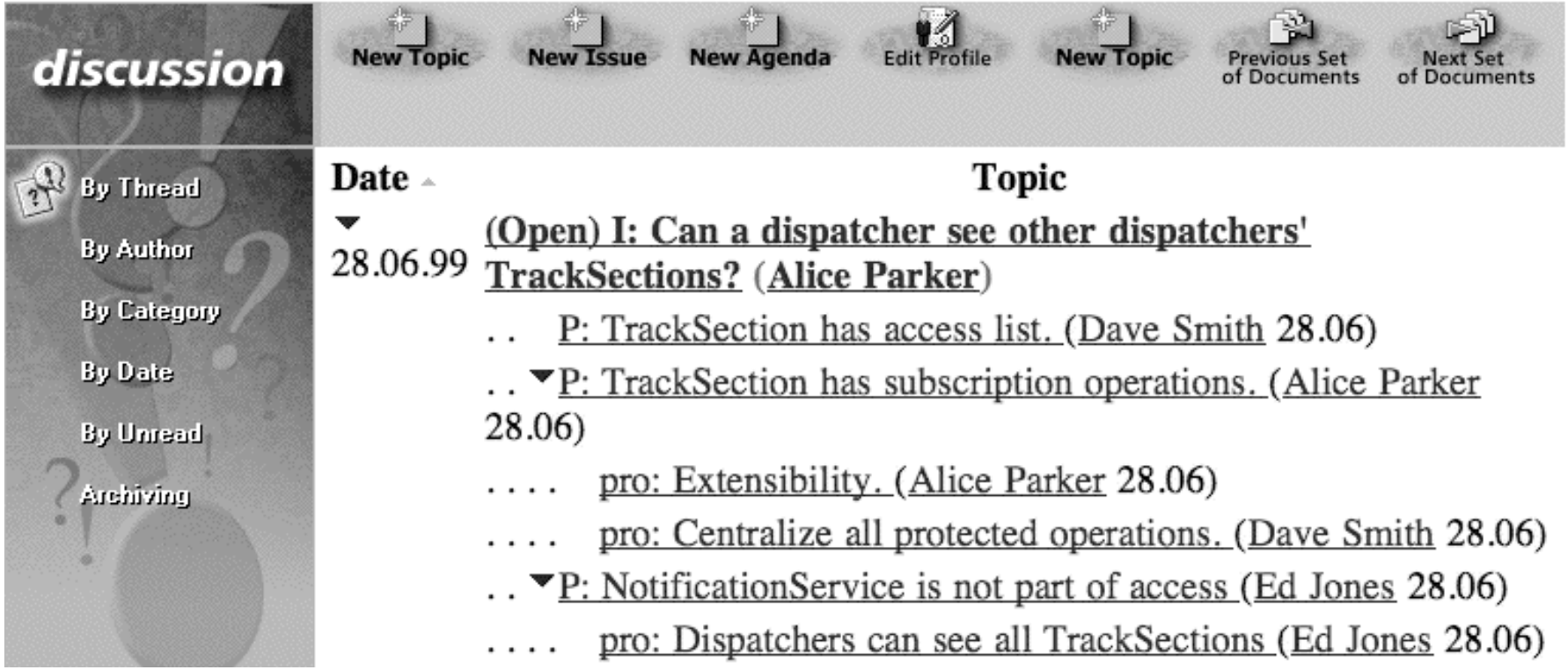
- A participant reports a problem and proposes a solution
- Change requests are often formalized when the project size is substantial
- Example: Request for additional functionality

```
Report number: 1291   Date: 5/3  Author: Dave
Synopsis: The STARS form should have a galaxy field.
Subsystem: Universe classification
Version: 3.4.1
Classification: missing functionality
Severity: severe
Proposed solution: …
```

Unplanned Communication Activities/Events

Issue resolution

- Selects a single solution to a problem for which several solutions have been proposed
- Uses issue base to collect problems and proposals.



Synchronous Communication Mechanisms

- Smoke signals
- Hallway conversation
  - Supports: Unplanned conversations, Request for clarification, request for change
  + Cheap and effective for resolving simple problems
  – Information loss, misunderstandings are frequent
- Meeting (face-to-face, phone, video conference)
  - Supports: Planned conversations, client review, project review, status review, brainstorming, issue resolution
  + Effective for issue resolution and consensus building
  – High cost (people, resources), low bandwidth.

Asynchronous Communication Mechanisms

- E-Mail
  - Supports: Release, change request, brainstorming
  + Ideal for planned communication and announcements
  – E-mail taken out of context can be misunderstood, sent to the wrong person, or lost
- Newsgroup
  - Supports: Release, change request, brainstorming
  + Suited for discussion among people who share a common interest; cheap (shareware available)
  – Primitive access control (often, you are either in or out)
- World Wide Web (Portal)
  - Supports: Release, change request, inspections
  + Provide the user with a hypertext metaphor: Documents contain links to other documents.
  – Does not easily support rapidly evolving documents.

Communication Activities

- Understand problem statement
- Join a team, schedule and attend team status meetings
  - Important task for the team leader:
    - Train the teams in meeting management

    Announce agendas

    Write minutes

    Keep track of action items
    - Show value of status meeting
    - Show time-saving improvements.
- Join the communication infrastructure.
  - A good communication infrastructure is the backbone of any software project
    - Learn to use the appropriate communication mechanism for the information at hand
    - Register for each communication mechanism which is used by the software project

    Get an account, get training

Summary

- Communication can be planned/unplanned, synchronous/asynchronous
- Communication is usually done via meetings, reviews, issues/change handling
- Online tools: Podio, Monday, ⋯ Think about overhead (getting in and out) and privacy
- Communication Questions to ask:
  - Are meetings scheduled in a calendar?
  - Does the project have a problem reporting system?
  - Do team members provide peer reviews in meetings or in written form?

Exercises

- 3-1: What is the difference between a role and a participant?
- 3-2: Can a role be shared between two or more pariticpants? Why or Why not?
- 3-7: What is the difference between a work package and a work product? Give examples of work packages and work products in producing a case management system like the Open Case Manager
- 3-8: What is the difference between a cross-functional team and a subsystem team?
- 3-11: As member of a user interface team developing a form you should find out which fields are required and which are optional. How do you find out?