Object-Oriented Software Engineering

Conquering Complex and Changing Systems

Outline

- Concepts and terminology
- Purpose of Software Project Management Plans
- Structure of a Project Management Plan
- Project responsibilities
- Team structures
- Project planning
- Work breakdown structure
- Communication Management
- Dependencies
- Schedule
- Project Management Tools

Software Project Management Plan

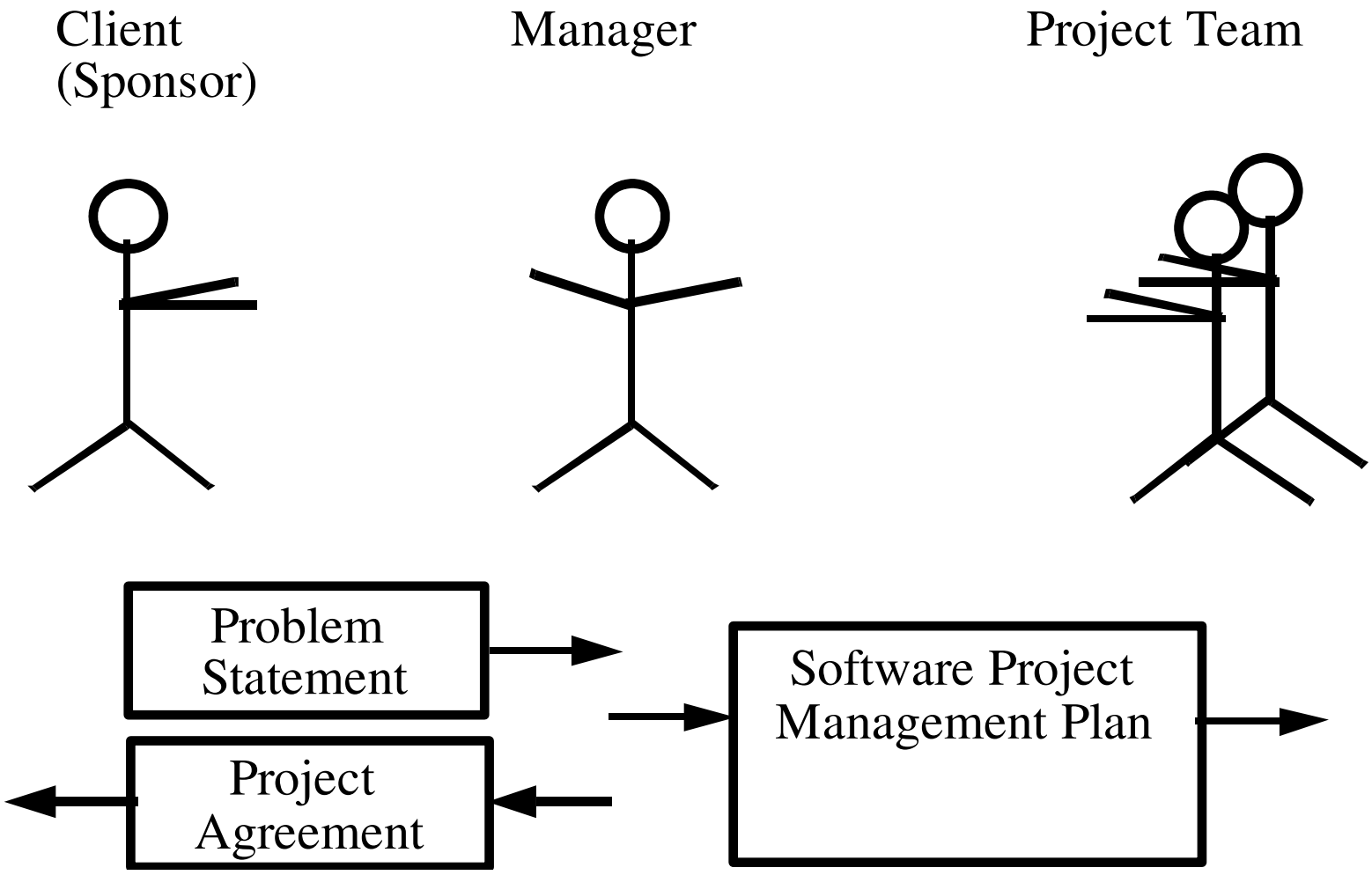
- Software Project:
  - **All *technical* and *managerial* activities required to deliver the deliverables to the client.**
  - **A software project has a specific duration, consumes resources and produces *work products*.**
  - **Management categories to complete a software project:**
    - **Tasks, 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 may imply changes in the other document.**
  - **SPMP may be part of project agreement.**

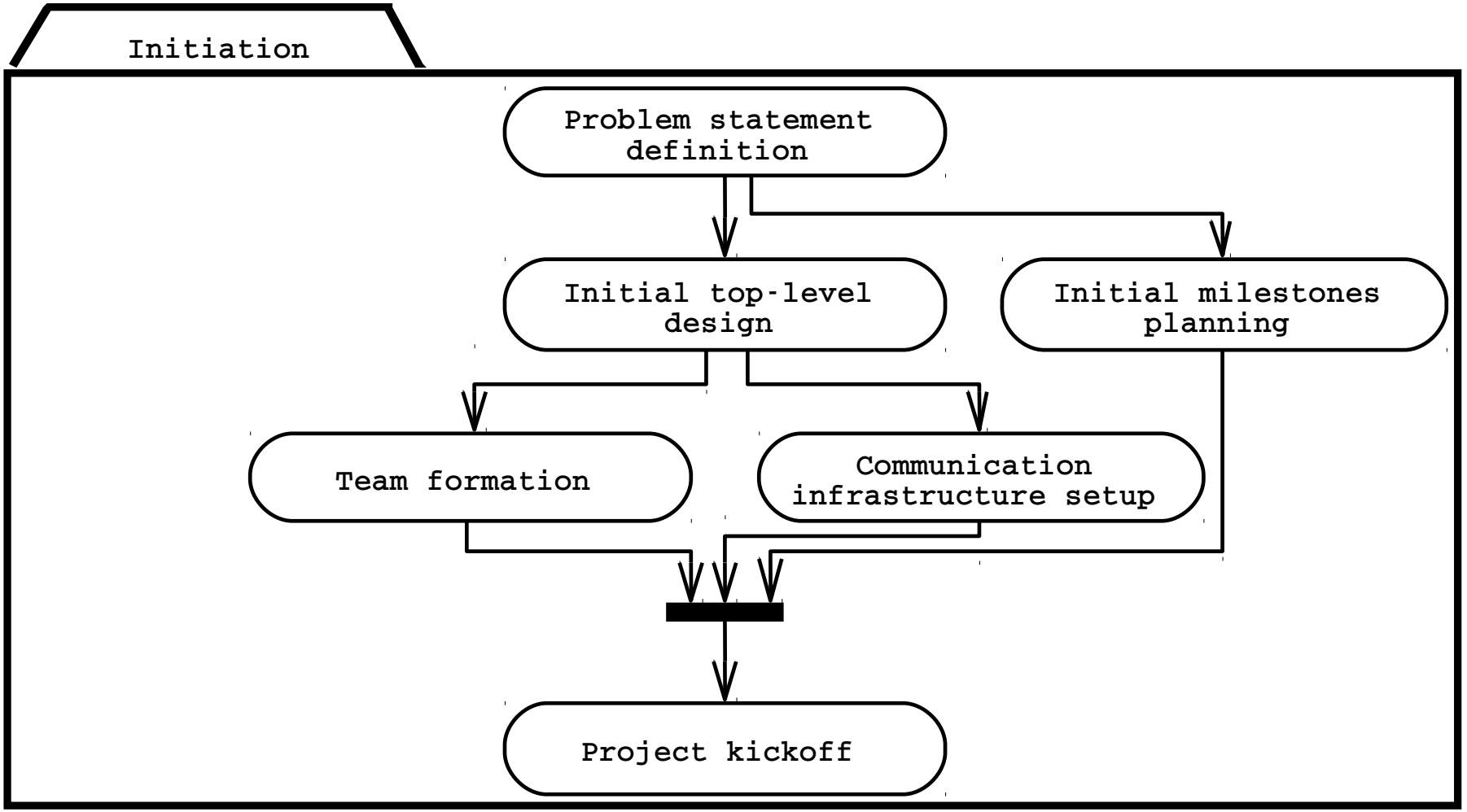
Project Agreement

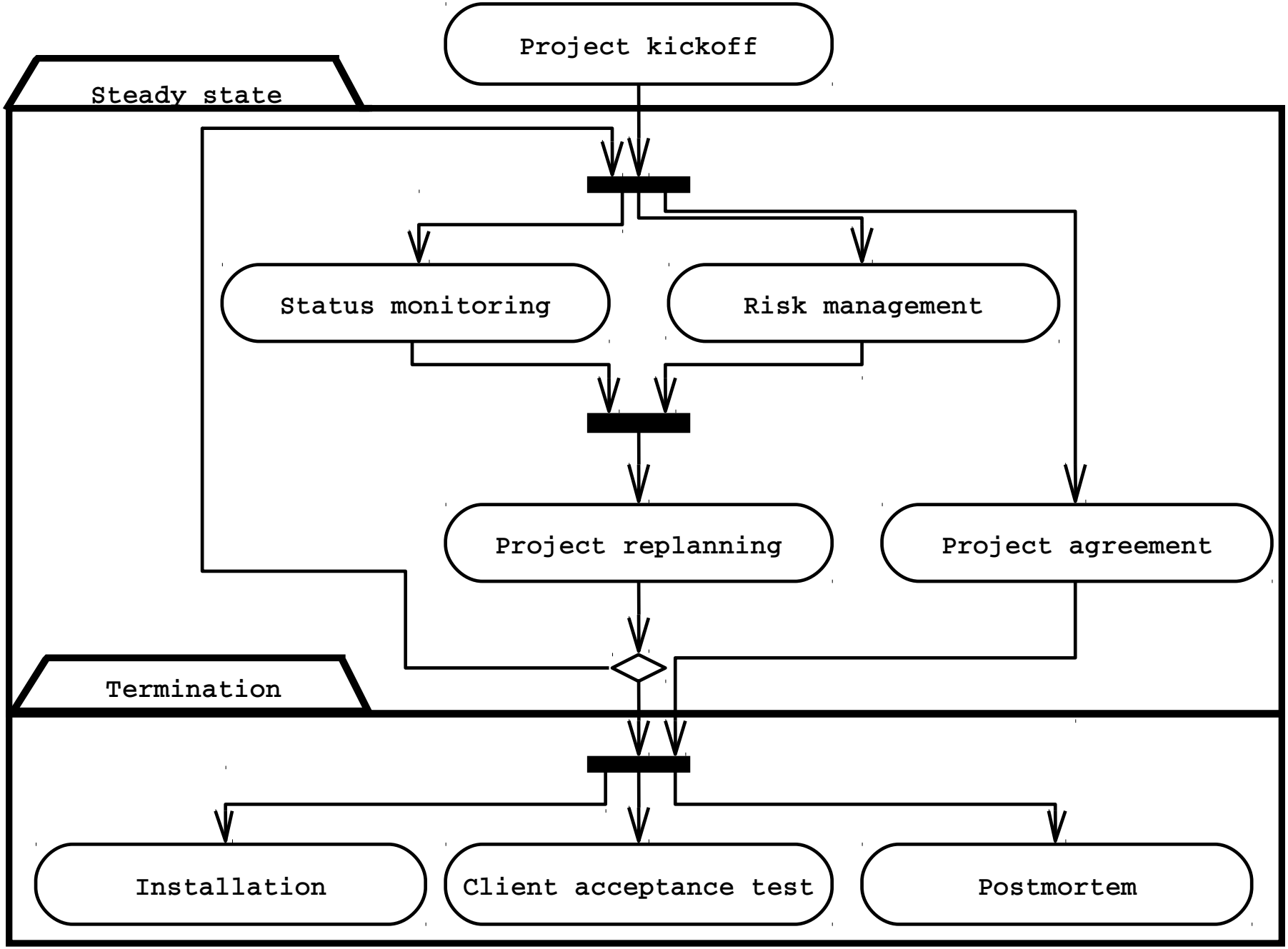
- Document written for a client that defines:
  - **the scope, duration, cost and deliverables for the project.**
  - **the exact items, quantities, delivery dates, delivery location.**
- Can be a contract, a statement of work, a business plan, or a project charter.
- Client: Individual or organization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

Project Agreement vs Problem Statement

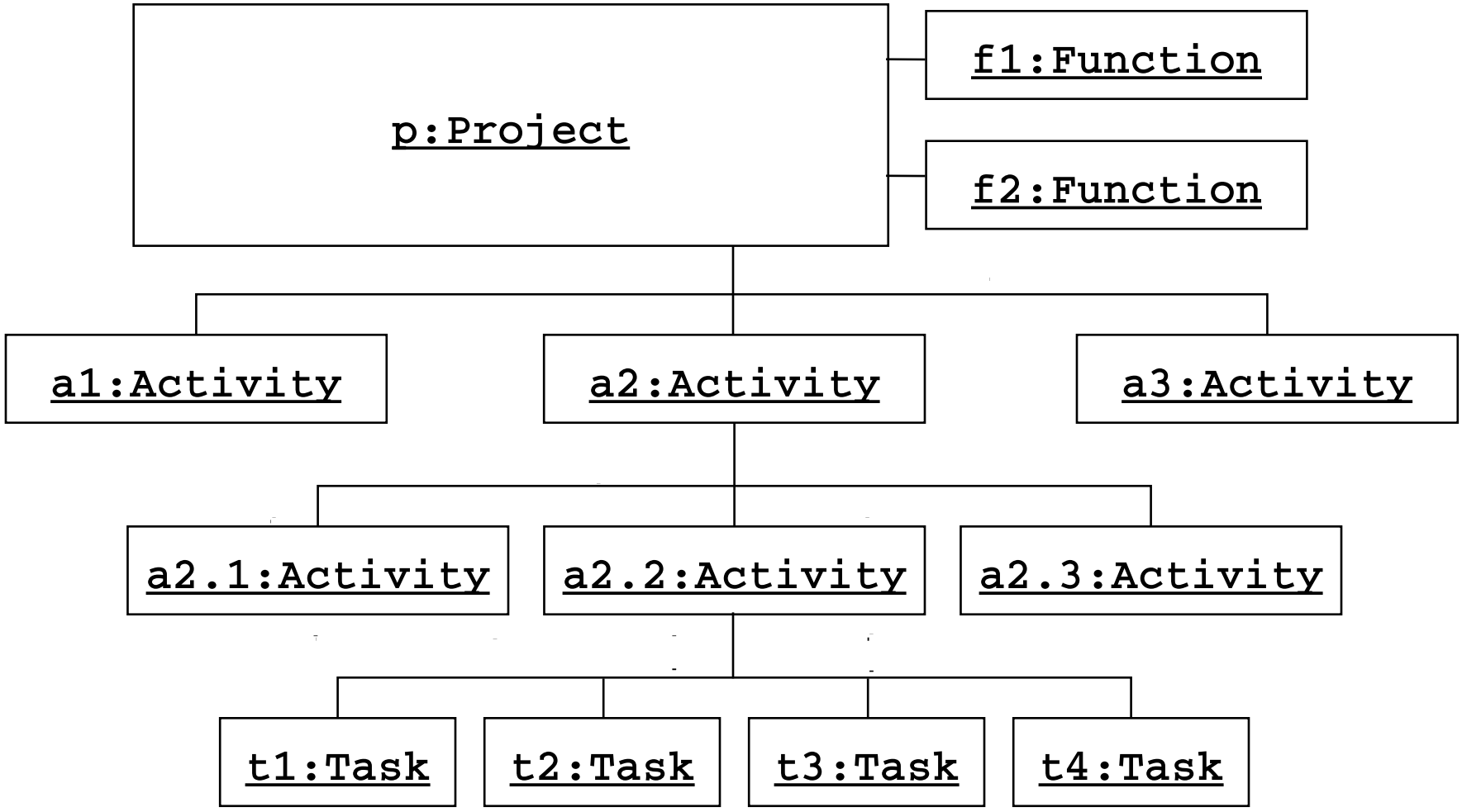


Project Management Activities (continued on next slide)



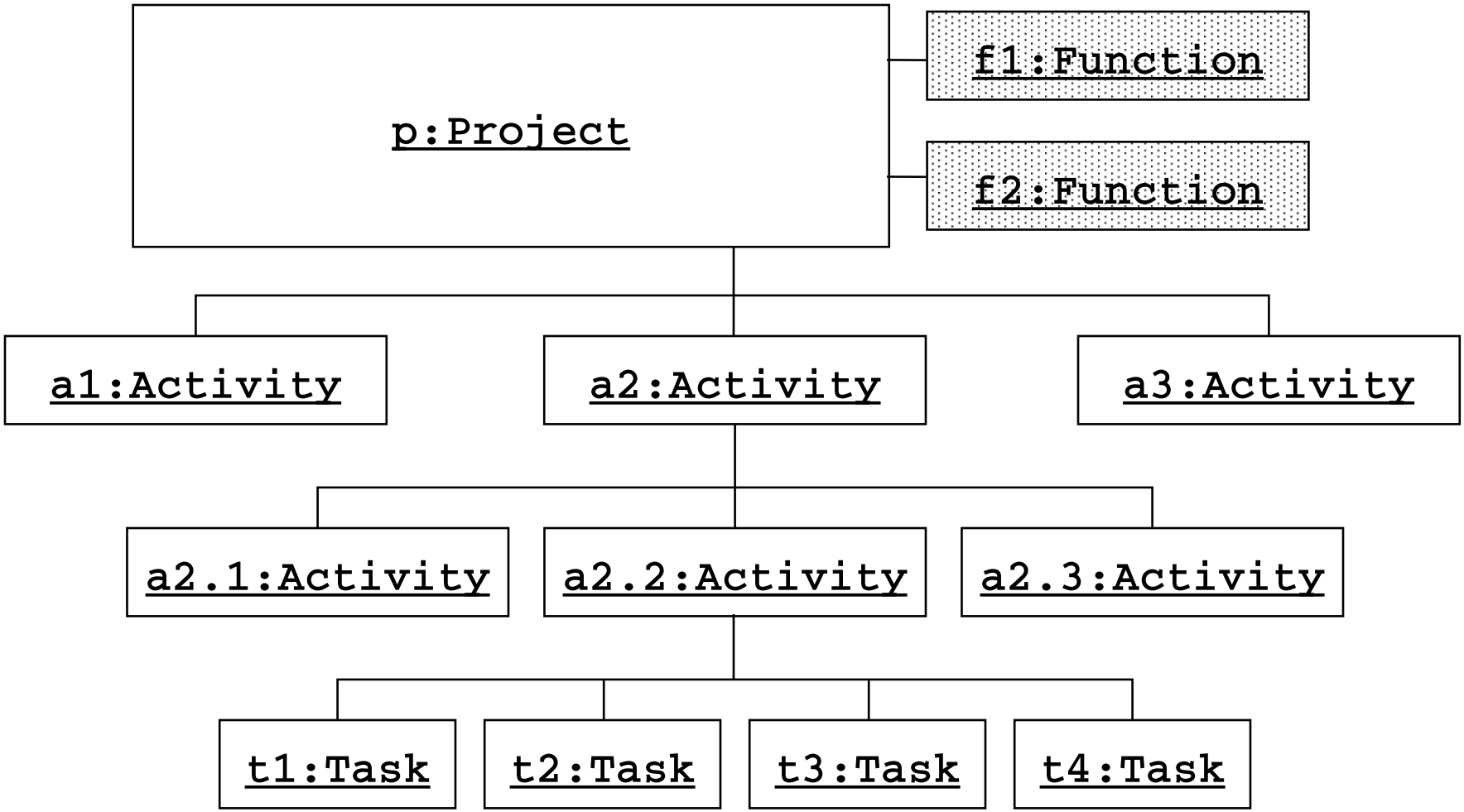

Project kickoff
Steady state
Status monitoring
Risk management
Project replanning
Project agreement
Termination
Installation
Client acceptance test
Postmortem

Project: Functions, Activities and Tasks



Functions

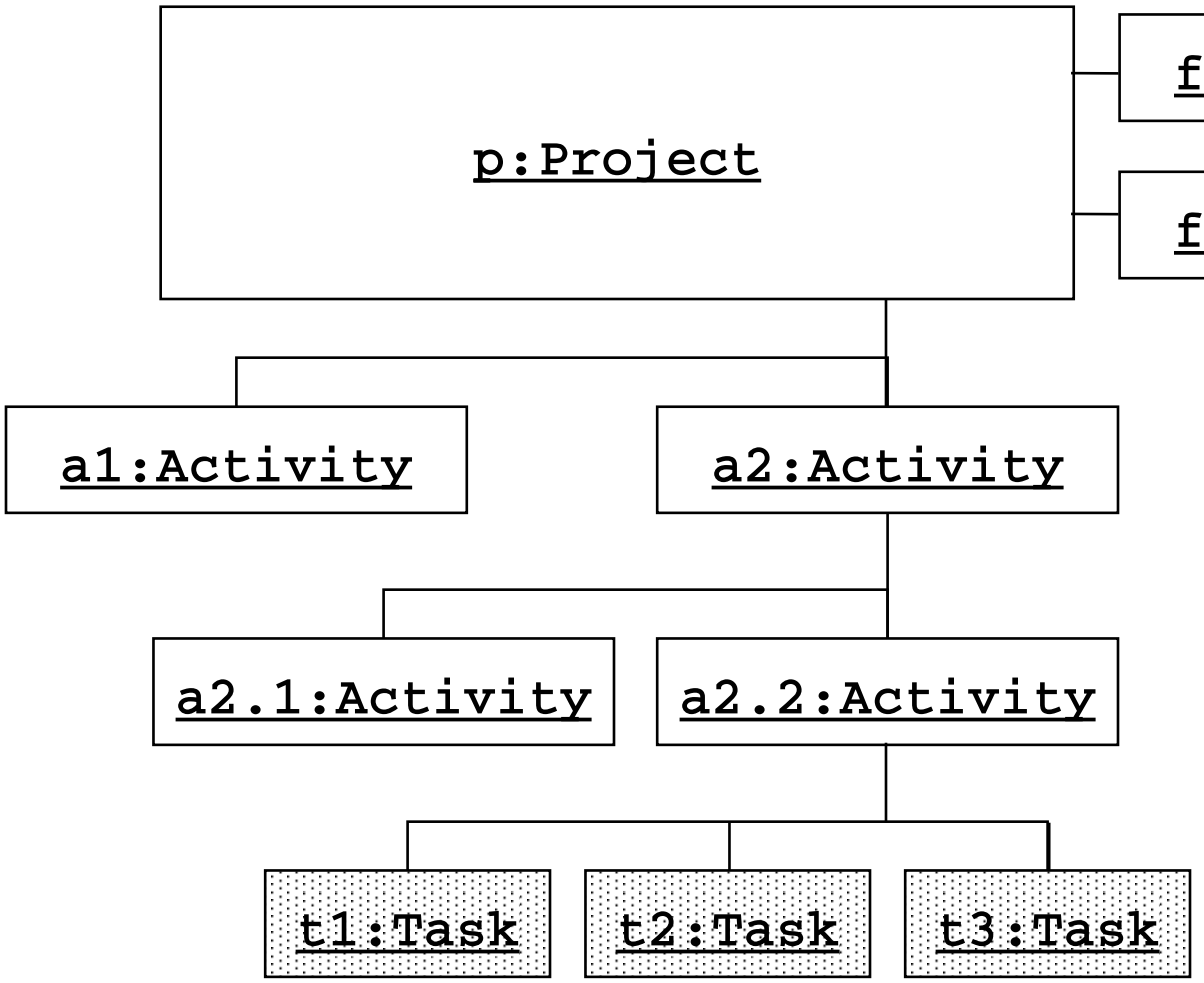
- Activity or set of activities that span the duration of the project



Functions

- Examples:
  - **Project management**
  - **Configuration Management**
  - **Documentation**
  - **Programming**
  - **Quality Control (Verification and validation)**
  - **Training**
- Question: Is system integration a project function?
- Mapping of terms: Project Functions in the IEEE 1058 standard are called **Integral processes** in the IEEE 1074 standard. We call them cross-development processes

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**
  - **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**
  - **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.**

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 activity identifies more tasks and modifies existing ones**
- Tasks must be decomposed into sizes that allow monitoring
  - **Work package usually corresponds to well defined work assignment for one worker for a week or a month.**
  - **Depends on nature of work and how well task is understood.**

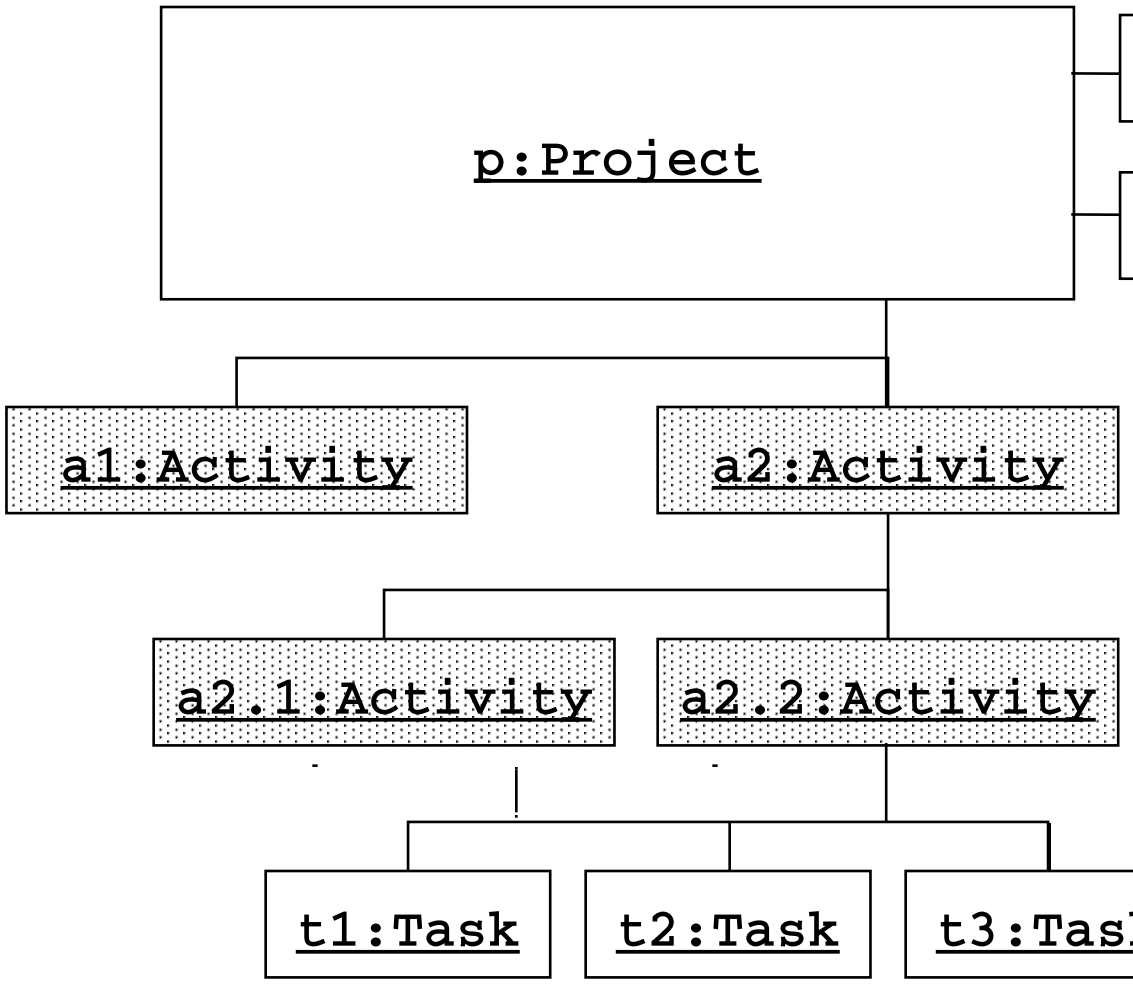
Examples of Tasks

- Unit test class “Foo”
- Test subsystem “Bla”
- Write user manual
- Write meeting minutes and post them
- Write a memo on NT vs Unix
- Schedule the code review
- Develop the project plan
- Related tasks are grouped into hierarchical sets of functions and activities.
- Action item

Action Item

- Definition: A task assigned to a person that has to be done within a week or less
- Action items
  - **Appear on the agenda in the Status Section (See lecture on communication)**
  - **Cover: What?, Who?, When?**
- Example of action items:
  - **Denise unit tests class "Foo" by next week**
  - **Marcus develops a project plan before the next meeting**
  - **Bob posts the next agenda for the Simulation team meeting before Sep 10, 12noon.**
  - **The team develops the project plan by Sep 18**

Activities



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

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 baseline:
  - **formally reviewed work product**
  - **under change control (change requires formal procedures)**
- Activities may be grouped into larger activities:
  - **Establishes hierarchical structure for project (phase, step, ...)**
  - **Allows separation of concerns**
  - **Precedence relations often exist among activities (PERT Chart)**

Examples of Activities

- Major Activities:
  - **Planning**
  - **Requirements Elicitation**
  - **Requirements Analysis**
  - **System Design**
  - **Object Design**
  - **Implementation**
  - **System Testing**
  - **Delivery**
- Activities during requirements analysis:
  - **Refine scenarios**
  - **Define Use Case model**
  - **Define object model**
  - **Define dynamic model**
  - **Design User Interface**

Structure of a Software Project Management Plan

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**

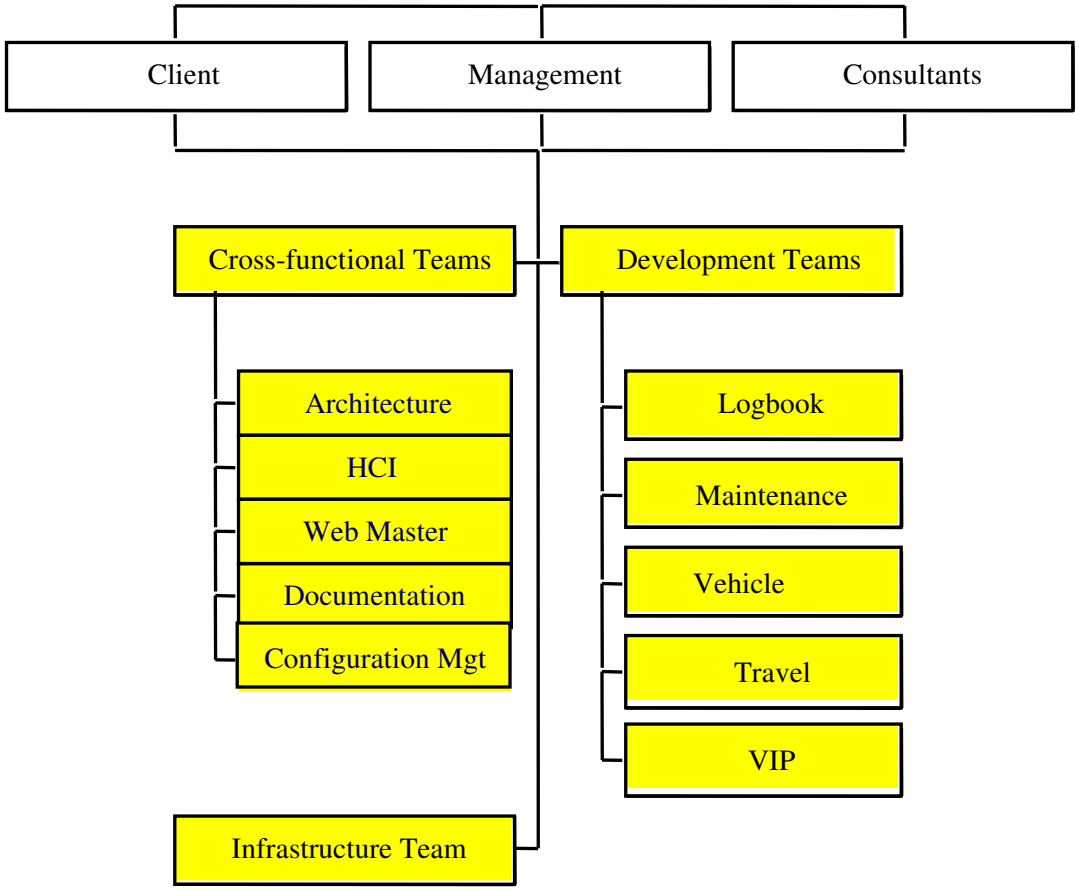
2.4 Project Responsibilities

- **Major functions and activities; nature of each; who's in charge**

Process Model

- Shows relationships among
  - **Functions, activities, tasks**
  - **Milestones**
  - **Baselines**
  - **Reviews**
  - **Work breakdown structure**
  - **Project deliverables**
  - **Sign-offs**
- Visualization of process model
- Project Management Aids
  - **MS Project (Microsoft)**
  - **MAC Project (Claris)**
  - **EasyTrak (Planning Control International)**

Example of an Organization Chart



SPMP Part 3: Managerial Processes

3.1 Management Objectives and Priorities

- **Philosophy, goals and priorities**

3.2 Assumptions, Dependencies, Constraints

- **External factors**

3.3 Risk Management

- **Identifying, assessing, tracking, contingencies for risks**

3.4 Monitoring and Controlling Mechanisms

- **Reporting mechanisms and formats, information flows, reviews**

3.5 Staffing Plan

- **Needed skills (what?, how much?, when?)**

Examples of Assumptions

- There are enough cycles on the development machines
- Security will not be addressed
- There are no bugs in Together-J, the CASE Tool recommended for the project
- A demonstration of the Starnetwork system will be given by the client

Examples of Dependencies

- The database team depends on the EPC database provided by DaimlerChrysler
- The automatic code generation facility in the CASE tool depends on JDK. The current release of Together-J supports only JDK 1.1.6

Examples of Constraints

- The length of the project is 3 months. limited amount of time to build the system
- The project consists of beginners. It will take time to learn how to use the tools
- Not every project member is always up-to-date with respect to the project status
- The use of UML and a CASE tool is required
- Any new code must be written in Java
- The system must use Java JDK 1.1.6

Risk Management

- **Risk: Members in key roles drop the course.**
  - **Contingency: Roles are assigned to somebody else. Functionality of the system is renegotiated with the client.**
- **Risk: The project is falling behind schedule.**
  - **Contingency: Extra project meetings are scheduled.**
- **Risk: One subsystem does not provide the functionality needed by another subsystem.**
  - ***Contingency: ?***
- **Risk: Ibutton runs only under JDK 1.2**
  - ***Contingency: ?***

SPMP Part 4: Technical Process

4.1 Methods, Tools and Techniques

- **Computing system, development method, team structure, etc.**
- **Standards, guidelines, policies.**

4.2 Software Documentation

- **Documentation plan, including milestones, reviews and baselines.**

4.3 Project Support Functions

- **Plans for functions (quality assurance, configuration management).**

SPMP Part 5: Work Elements

5.1 Work Packages (Work breakdown structure)

- **Project decomposed into tasks; definitions of tasks**

5.2 Dependencies

- **Precedence relations among functions, activities and tasks**

5.3 Resource Requirements

- **Estimates for resources such as personnel, computer time, special hardware, support software.**

5.4 Budget and Resource Allocation

- **Connect costs to functions, activities and tasks.**

5.5 Schedule

- **Deadlines, accounting for dependencies, required milestones**

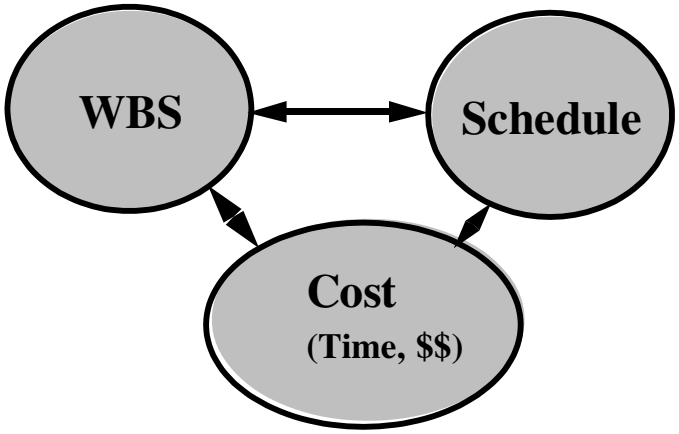
Creating Work Packages

- Work Breakdown Structure (WBS) (Section 5.1)
  - **Break up project into activities (phases, steps) and tasks.**
  - **The work breakdown structure does not show the interdependence of the tasks**

- The identification of the work breakdown structure is an instance of object identification and associating these objects

WBS Trade-offs

- Work breakdown structure influences cost and schedule
- Thresholds for establishing 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**
- Determination of work breakdown structure is incremental and iterative



Source: Software Engineering Economics, Barry W. Boehm p. 47, Prentice Hall, N.J., 1981

Dependencies and Schedule (SPMP Section 5.2 + 5.5)

- An important temporal relation: "must be preceded by"
- Dependency graphs show dependencies of the tasks (hierarchical and temporal)
  - **Activity Graph:**
    - **Nodes of the graph are the project milestones**
    - **Lines linking the nodes represent the tasks involved**
  - **Schedule Chart (MS-Project):**
    - **Nodes are tasks and milestones**
    - **Lines represent temporal dependencies**
- Estimate the duration of each task
- Label dependency graph with the estimates

Project Management Tools for Work Packages

- Visualization Aids for Project Presentation
  - **Graphs (Schedule), Trees (WBS)**
  - **Tables (Resources)**
- Task Timeline
  - **Gantt Charts: Shows project activities and tasks in parallel. Enables the project manager to understand which tasks can be performed concurrently.**
- Schedule Chart (PERT Chart)
  - **Cornerstone in many project management tools**
  - **Graphically shows dependencies of tasks and milestones**
    - **PERT: Program Evaluation and Review Technique**
      - **A PERT chart assumes normal distribution of tasks durations**
      - **Useful for Critical Path Analysis**
    - **CPM: Critical Path Method**

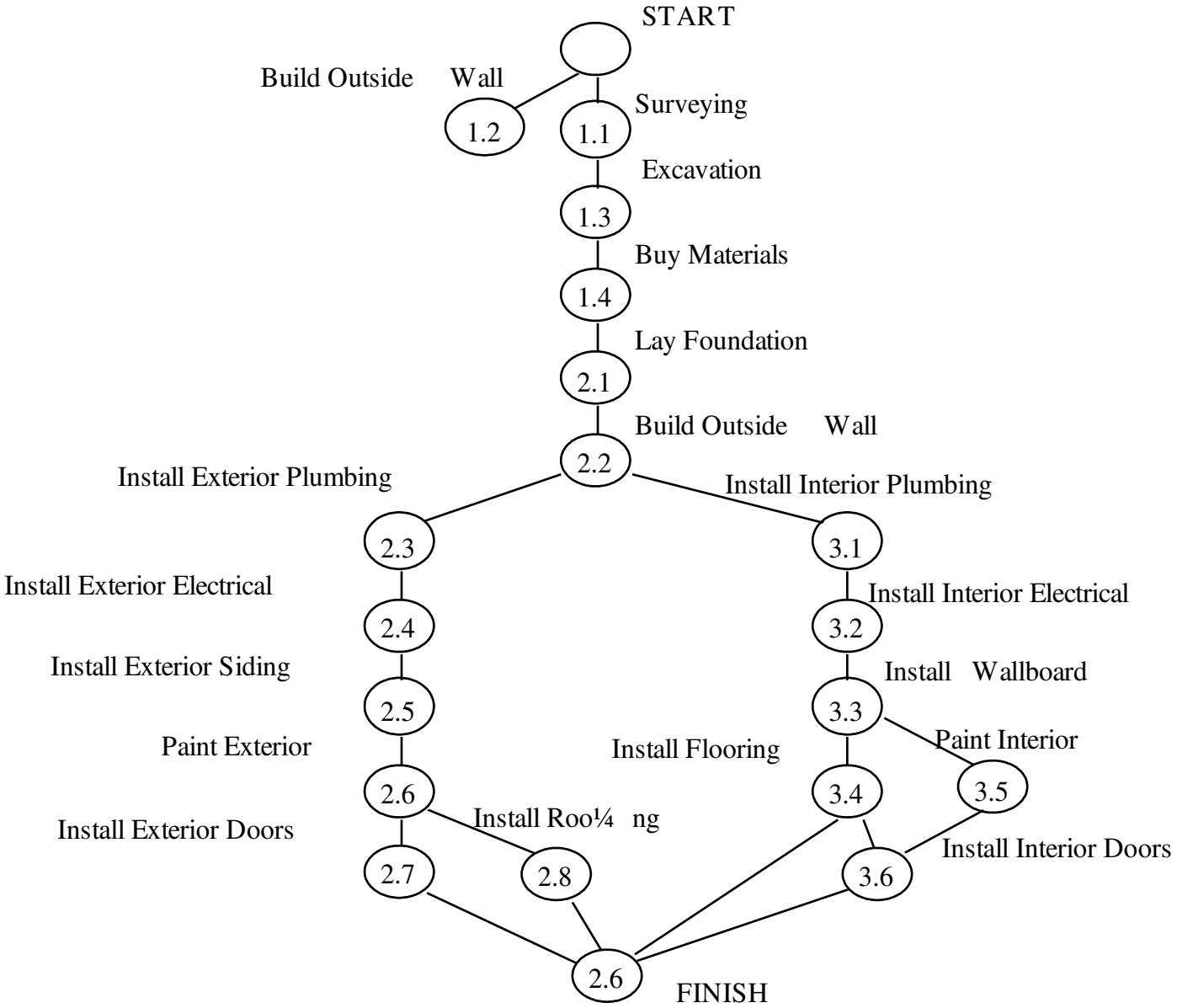
Project: Building a House

- Activity 1: Landscaping the lot
  - **Task 1.1: Clearing and grubbing**
  - **Task 1.2: Seeding the Turf**
  - **Task 1.3: Planting shrubs and trees**
- Activity 2: Building the House
  - **Activity 2.1 : Site preparation**
  - **Activity 2.2: Building the exterior**
  - **Activity 2.3: Finishing the interior**
- Activity 2.1 : Site preparation
  - **Task 2.1.1: Surveying**
  - **Task 2.1.2: Obtaining permits**
  - **Task 2.1.3: Excavating**
- Task 2.1.4: Obtaining materials

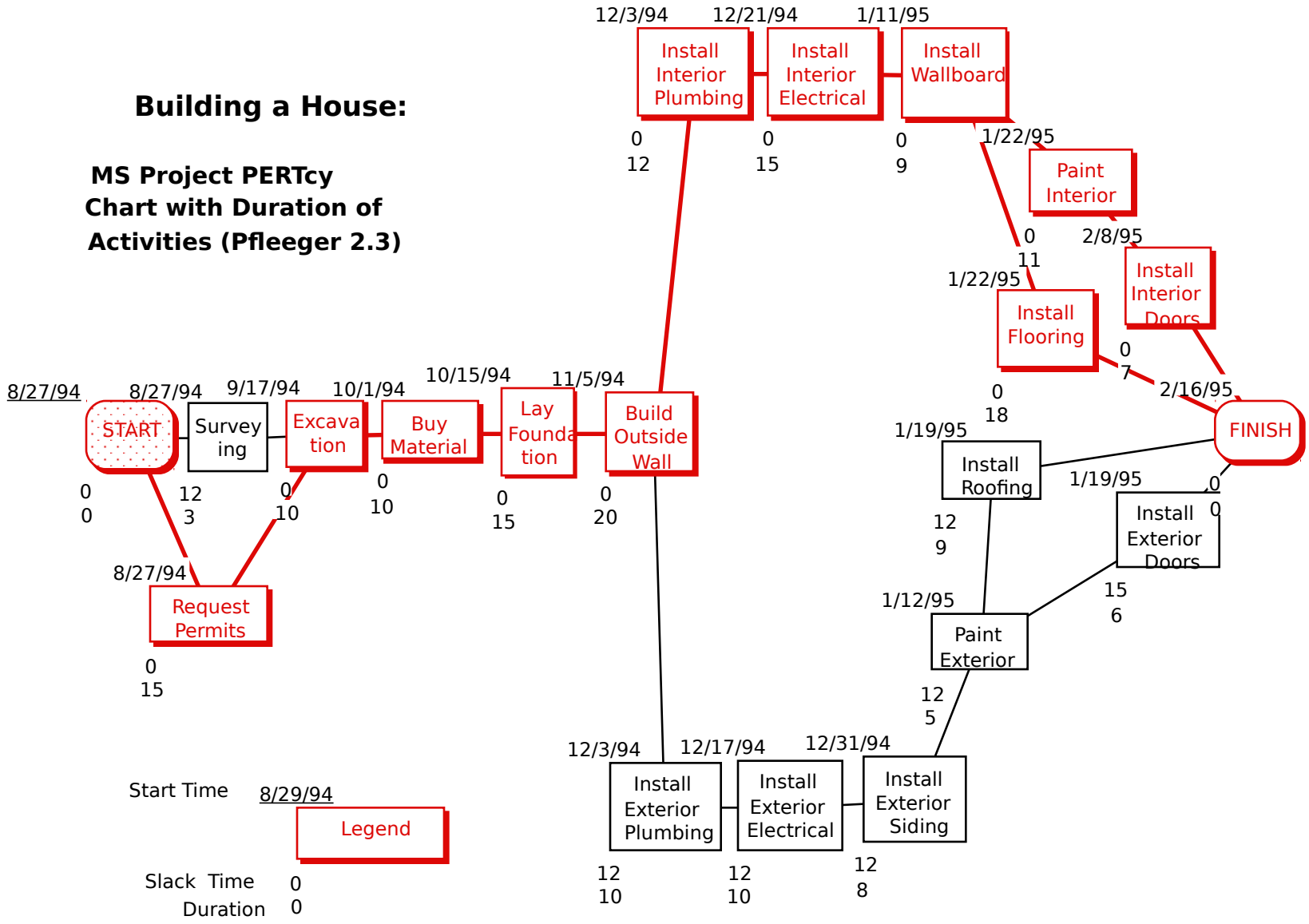
Activity 2: Building a House, ctd

- Activity 2.2: Building the exterior
  - **Task 2.2.1: Foundation**
  - **Task 2.2.2: Outside Walls**
  - **Task 2.2.3: Exterior plumbing**
  - **Task 2.2.4: Exterior electrical work**
  - **Task 2.2.5: Exterior siding**
  - **Task 2.2.6: Exterior painting**
  - **Task 2.2.7: Doors and Fixtures**
  - **Task 2.2.8: Roof**
- Activity 2.3 : Finishing the Interior
  - **Task 2.3.1: Interior plumbing**
  - **Task 2.3.2: Interior electrical work**
  - **Task 2.3.3: Wallboard**
  - **Task 2.3.4: Interior painting**
  - **Task 2.3.5: Floor covering**
  - **Task 2.3.6: Doors and fixtures**

Activity Graph for Activity "Building a House"



PERT Chart Example for "Building a House"



How do you become a good project planner?

- Establish a project plan
  - **Start with the plan based on your experience with the last project(s)**
- Keep track of activities and their duration
- Determine difference between planned and actual performance
- Make sure to do a post-mortem
  - **Lessons learned**
  - **Ask developers for feedback**
  - **Write a document about what could have been improved**

Project Management Heuristics

- Make sure to be able to revise or dump a project plan
  - **Complex system development is a nonlinear activity**
- If project goals are unclear and complex use team-based project management. In this case
  - **Avoid GANTT charts and PERT charts for projects with changing requirements**
  - **Don't look too far into the future**
- Avoid micro management of details
- Don't be surprise if current project management tools don't work:
  - **They were designed for projects with clear goals and fixed organizational structures**

Project Management Summary

- Get agreement among customers, managers and teams
  - **Problem statement**
  - **Software project management plan**
  - **Project agreement**
  - **Make sure agreement allows for iteration**
- Organization Structures
- SPMP
- Project planning
  - **Start with work breakdown structure (WBS)**
  - **Identify dependencies and structure: Tasks, activities, functions**
- Tools and Techniques
  - **GANTT, Dependency graph, Schedule, Critical Path Analysis**
  - **Be careful with tools in projects with a lot of change**

Varie:

- Esempio di documento: BOTS_SPMP.pdf

BackCasting (vs forward):

- Parto dal risultato finale desiderato alla data prevista e calcolo tutto cio' che e' necessario per la sua realizzazione, tornando via via indietro nel tempo
- Se la data di inizio e' molto prima di oggi, il progetto richiede piu' risorse di quello che credevamo
- Se e' molto dopo oggi, il progetto e' piu' facile da realizzare del previsto
- Vicino ad oggi: previsioni sensate

Vantaggi del BackCasting :

- Ogni task deriva dal fare parte di un risultato atteso, se ne conosce l'utilita', non si sprecano energie in task inutili
- Mancanze di risorse , conoscenza, tecnologie sono identificate all'inizio e non dopo, riducendo fortemnente I rischi associati
- Aree critiche o di aspettative irrealistiche sono individuate precocemente