

TUM

Work Breakdown Structures

16 May 2006

Bernd Bruegge

Technische Universität München

Institut für Informatik

Lehrstuhl für Angewandte Softwaretechnik

<http://www.bruegge.in.tum.de>

Where are we?

- ❖ In the last lecture we focused on general software project management issues and configuration management
 - ◆ How do we structure a project? How do we deal with change?
 - ◆ We introduced the SPMP and SCMP
- ❖ We now focus on specific software management issues
 - ◆ **Decomposition of work (WBS)**
 - ◆ What are the units of tasks that need to be done?
 - ◆ **Project Estimation**
 - ◆ What resources (people, \$) do we need for each of these tasks?
 - ◆ **Project Organization**
 - ◆ Who is doing these tasks?
 - ◆ **Scheduling**
 - ◆ How long does it take to finish them?

Schedule for next 4 Classes

- ◆ **Today' lecture:** Decomposition of work
- ◆ **Tomorrow's exercise:**
 - ◆ WBS Exercise (Advanced Home Dispatch Project)
 - ◆ Project Manager: Johannes Bross, Accenture
- ◆ **May 23-24**
 - ◆ Lecture and Exercise on estimation
- ◆ **May 30-31**
 - ◆ Lecture and Exercise on project organization
- ◆ **June 7-8**
 - ◆ **No lecture and no exercise**
- ◆ **June 13-14**
 - ◆ Lecture and Exercise on scheduling

Outline of Today's Lecture

- ❖ Handout for tomorrow's exercise
- ❖ Determining Work and Tasks Sizes
- ❖ Work Breakdown Structure (WBS)
- ❖ Different Approaches for developing WBSs
- ❖ Notations for Work Breakdown Structures
- ❖ Heuristics and examples for WBS
 - ◆ Starting with templates
 - ◆ How to identify work
 - ◆ What do you do with risky tasks?
- ❖ Using WBS in large projects
 - ◆ How detailed should a WBS be?
 - ◆ How can you plan the tasks of a long project when things are unknown or changing all the time?

Tomorrow's Exercise

- ❖ **Advanced Developer Telecommuting Project**
 - ◆ **Increase developer productivity by allowing them to work at home.**
 - ◆ **Objectives**
 - ◆ **Developers receive orders via a work assignment system**
 - ◆ **Develop a new reporting mechanism for this process**
 - ◆ **Train developers and supervisors for this process.**
- ❖ **2 Handouts**
 - ◆ **Problem Statement from client**
 - ◆ **Project Agreement from client**

Problem Statement: Advanced Developer Telecommuting Project

❖ **Background**

- ◆ Arena Services recently merged with Asteroids Games

❖ **Project**

- ◆ Advanced Developer Telecommuting project

❖ **Goal**

- ◆ Increase developer productivity and reduce overhead costs

❖ **Objectives**

- ◆ Complete project within budget, provide training materials
- ◆ Establish procedures for managing telecommuting developers, tracking laptop usage, how to run weekly meetings
- ◆ Produce training materials for task assignment workflow, work product delivery, compliance reporting

❖ **Constraints**

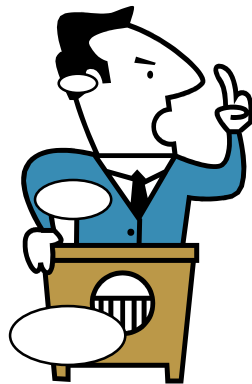
- ◆ Procedures and materials must comply with legal laws and labor relations

Tomorrow's Exercise: Project Agreement

- ❖ Project Description
- ❖ Key Tasks
- ❖ Job Roles and Titles
- ❖ Estimated Schedule for Delivery
- ❖ Deliverables
- ❖ High level Work Segments (“Work breakdown structure”)
 - ◆ Dependencies
- ❖ Assumptions
- ❖ Budget
- ❖ Invoices and Billing Information

What is the problem?

- ❖ Your boss: “How long will this take?”
- ❖ You: “Between 1 and 6 months.”



“As long as I can do it within 6 months, I keep my promise.”

“With hard work, he can do it in 1 month.”

What is the problem?

- ❖ Your boss: “How long will this take?”
- ❖ You: “Between 1 and 6 months.”



“I have not the slightest clue, if it is possible at all.”

“Even if it is possible, I don’t know, how long it will take.”

- ❖ Solution: Use divide and conquer
 - ◆ To give a good answer you break the work down into activities for which you try to get timing estimates
 - ◆ Only if you can get good estimates can you compute the estimated project duration.

Activities to obtain good time estimates

- ❖ Identify the work that needs to be done
 - ◆ **Work breakdown structure (WBS), SPMP Section 5.1**
- ❖ Identify the dependency between work units
 - ◆ **Dependency Graph, SPMP Section 5.2**
- ❖ Estimate the duration of the work to be done
 - ◆ **Schedule, SPMP Section 5.5.**

Software Project Management Plan

- ❖ 0. Front Matter
- ❖ 1. Introduction
- ❖ 2. Project Organization (Lecture on May 30)
- ❖ 3. Managerial Process
- ❖ 4. Technical Process
- ❖ 5. Work Elements, Schedule, Budget
 - ◆ **5.1 Work Breakdown Structure (WBS) (Today)**
 - ◆ **5.2 Dependencies between tasks (Today)**
 - ◆ **5.3 Resource Requirements (Lecture on May 23)**
 - ◆ **5.4 Budget (Lecture on May 23)**
 - ◆ **5.5 Schedule (Lecture on June 13)**
- ❖ Optional Inclusions

Software Project Management Plan

- ❖ 0. Front Matter
- ❖ 1. Introduction
- 2. Project Organization (Lecture on May 30)
- ❖ 3. Managerial Process
- ❖ 4. Technical Process
- ❖ 5. Work Elements, Schedule, Budget
 - 5.1 Work Breakdown Structure (WBS) (Today)
 - 5.2 Dependencies between tasks (Today)
 - 5.3 Resource Requirements (Lecture on May 23)
 - 5.4 Budget (Lecture on May 23)
 - 5.5 Schedule (Lecture on June 13)
- ❖ Optional Inclusions

Let's Build a House

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

First Step: Identify the work to be done

- ❖ 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 tasks initially is a brainstorming activity

Similar to activities used during requirements elicitation and analysis.

Second Step: Hierarchically organize the Work

- ❖ 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

Activity

Sub-Activity or Task

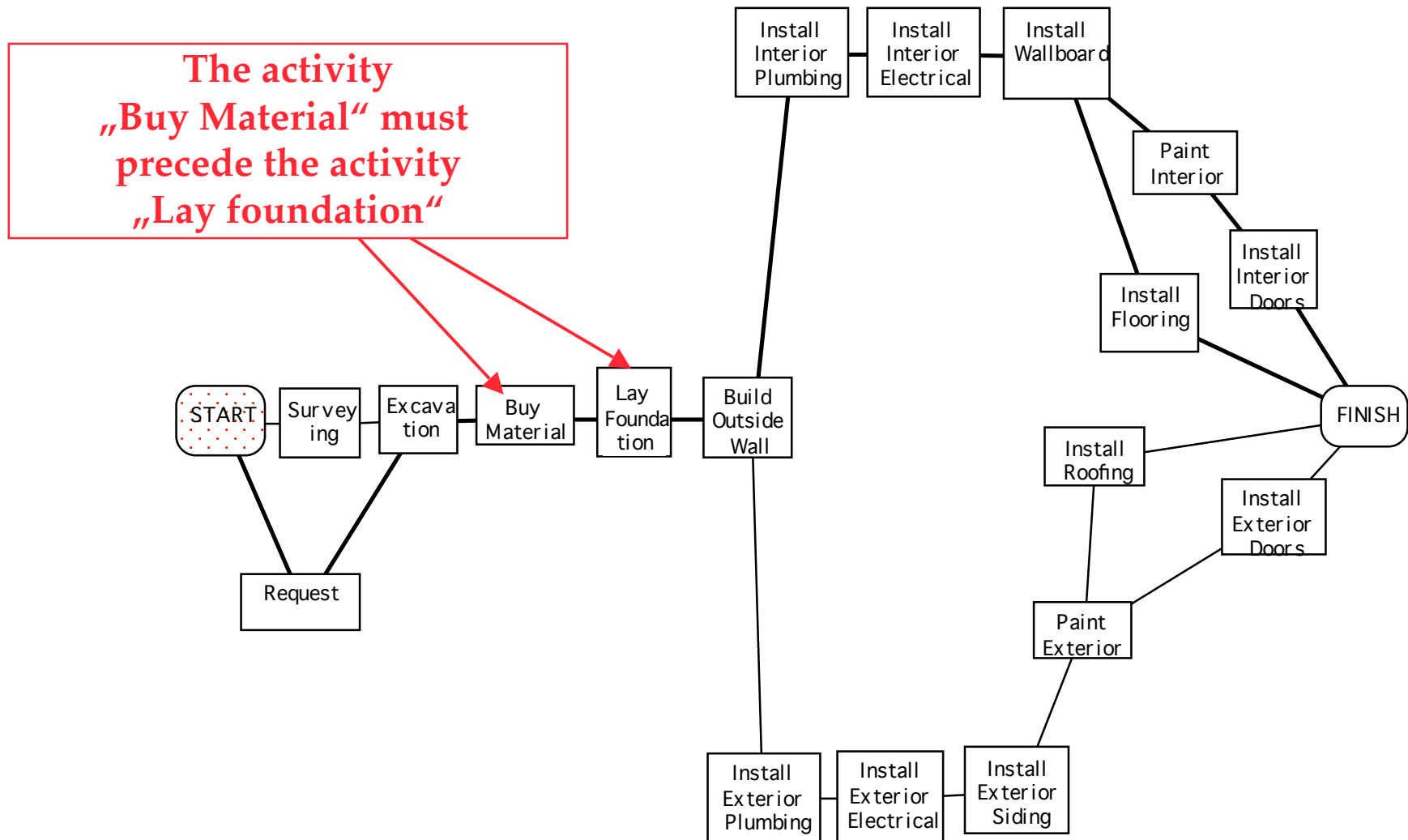
Finding this hierarchy involves categorization and refinement.

Good after brainstorming, not during brainstorming

Third Step: Identify Dependencies between Work

- ❖ **The work breakdown structure does not show any dependence among activities/tasks**
 - ◆ Can we excavate before getting the permit?
 - ◆ How much time do I need for the whole project, if I know the individual times?
 - ◆ What can be done in parallel?
 - ◆ Are there any critical activities, that can significantly slow down the project?
- ❖ **Dependencies like these are shown in the dependency graph**
 - ◆ Nodes are activities
 - ◆ Lines represent temporal dependencies.

Building a House (Dependency Graph)

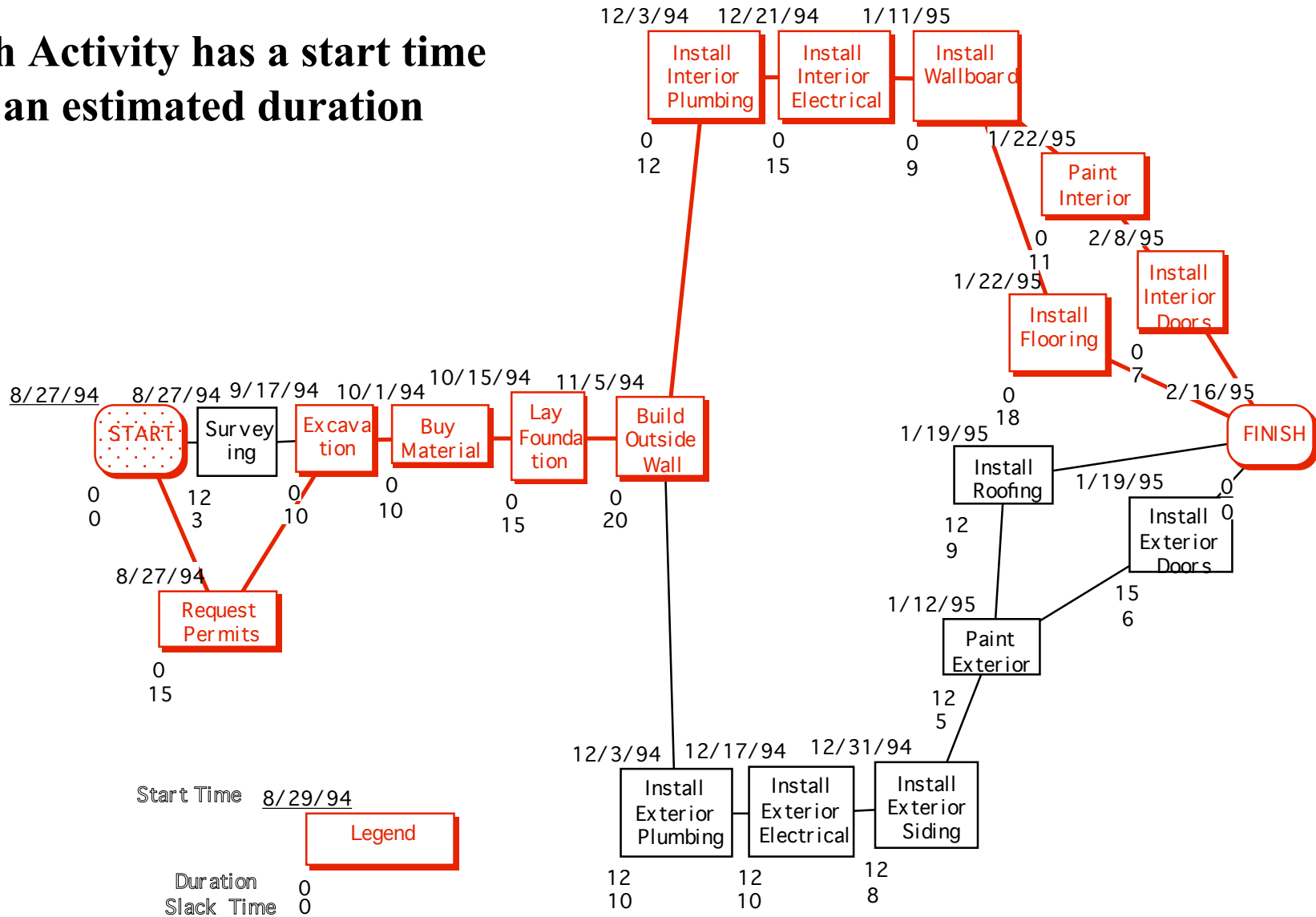


Fourth step: Map the Work onto Time

- ❖ **Estimate starting times and durations for each of the tasks in the dependency graph**
- ❖ **Compute the longest path through the graph: This is the estimated duration of your project.**

Building a House (Schedule, PERT Chart)

Each Activity has a start time and an estimated duration



How do we get good Estimates?

- ❖ **Estimation of starting times and durations is crucial for setting up a plan**
- ❖ **In the lecture on scheduling we will discuss methods and heuristics on how to do it and how to establish a software project schedule**
- ❖ **First let us learn a few more technical terms defined in the SPMP IEEE Std 1058.**

Recall Definitions from Lecture 1

- ❖ **Project:**
 - ◆ A Project has a duration and consists of functions, activities and tasks
- ❖ **Work Package:**
 - ◆ A description of 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
 - ◆ can only be changed through a formal change procedure
- ❖ **Project Deliverable:**
 - ◆ A work product to be delivered to the customer.

Activities, Tasks and Functions

- ❖ **Activity:** A a major unit of work with precise dates that consists of smaller activities or tasks
 - ◆ Culminates in a project milestone
- ❖ **Task:** Smallest unit of work subject to management
 - ◆ Small enough for adequate planning and tracking
 - ◆ Large enough to avoid micro management
- ❖ **Project Function:** An activity or set of activities that span the duration of the project.

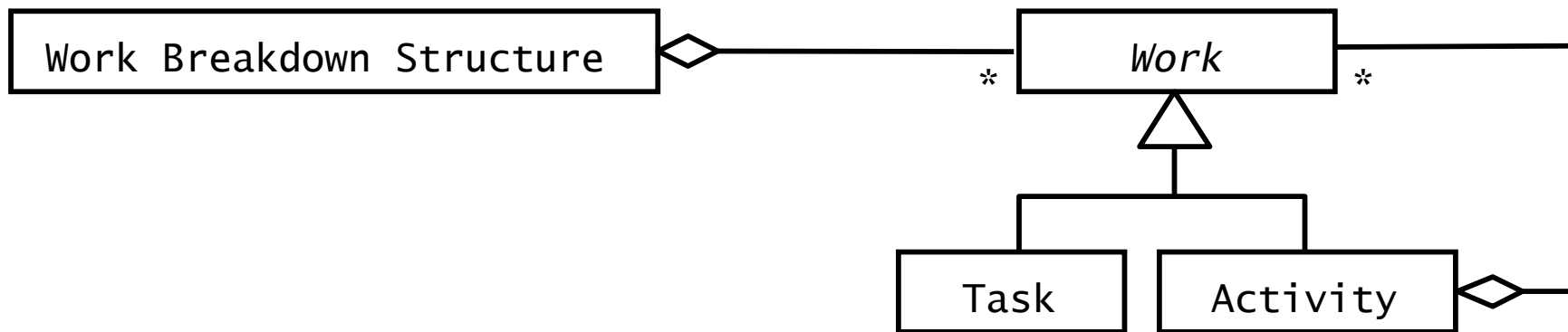
Tasks

- ❖ Smallest unit of management accountability
 - ◆ Atomic unit of planning and tracking
 - ◆ Tasks have finite duration, need resources, produce tangible result (documents, code)
- ❖ The **description of a task** is done in a **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 products 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 and templates 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
 - ◆ Depends on nature of work and how well task is understood
 - ◆ Work package usually corresponds to a well defined work assignment for one worker for a week
 - ◆ Also called action item.

Work Breakdown Structure



Work Breakdown Structure: The aggregation of all the work to be performed in a project. Often called **WBS**.

Approaches to Develop Work Breakdown Structures

- ❖ **Product component approach** 
 - ◆ Structure the work based on the work products
 - ◆ Examples: Design documents, manuals, delivered system
- ❖ **Functional approach**
 - ◆ Structure the work based on development activities and project functions
 - ◆ Examples: Analysis, design, implementation, integration
- ❖ **Geographical area approach**
 - ◆ Structure the work based on geographical location
 - ◆ Examples: Munich team, Pittsburgh team, off-shore team
- ❖ **Organizational approach**
 - ◆ Structure the work based on the organizational structure
 - ◆ Example: R&D department, predevelopment, product development, marketing, sales.

When to use what Approach

- ❖ The teams are distributed over the continent:
 - ◆ **Geographical area approach**
- ❖ The teams consist of experienced developers:
 - ◆ **Product component approach**
- ❖ The project has mostly beginners or an unexperienced project manager:
 - ◆ **Functional approach**
- ❖ The project is a continuation of a previously successful project, there are no changes in the requirements and no new technology enablers
 - ◆ **Organizational approach**

Whatever approach you choose, stick with it to prevent possible overlap in categories.

Mixing different Approaches is bad

- ❖ Consider the WBS for the activity „Prepare report“
- ❖ WBS using the functional approach:
 - ◆ Write draft report
 - ◆ Have draft report reviewed
 - ◆ Write final report
- ❖ WBS using the product component approach:
 - ◆ Chapter 1
 - ◆ Chapter 2
 - ◆ Chapter 3
- ❖ WBS using a mixed approach:
 - ◆ Chapter 1
 - ◆ Chapter 2
 - ◆ Chapter 3
 - ◆ Have draft report reviewed
 - ◆ Write final report



Why is this bad?

How do you develop a good WBS?

- ❖ **Top down approach:**
 - ◆ Start at the highest, top level activities and systematically develop increasing levels of detail for all activities
- ❖ **Bottom up approach (“Brainstorming”):**
 - ◆ Generate all activities you can think of that will have to be done and then group them into categories
- ❖ **Which one you use depends on**
 - ◆ how familiar you and your team are with the project,
 - ◆ whether similar projects have successfully been performed in the past, and
 - ◆ how many new methods and technologies will be used.

The Top Down WBS Development

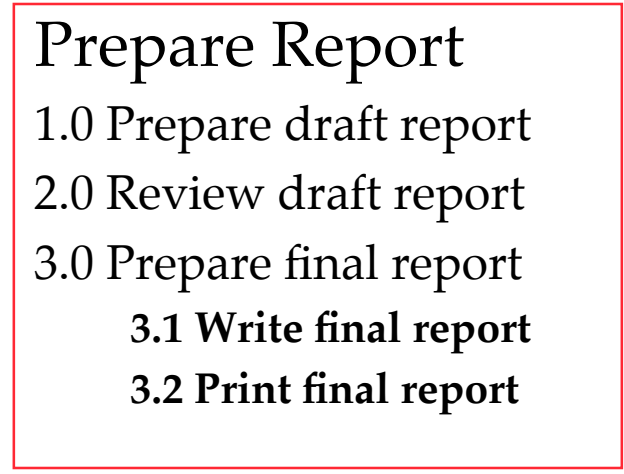
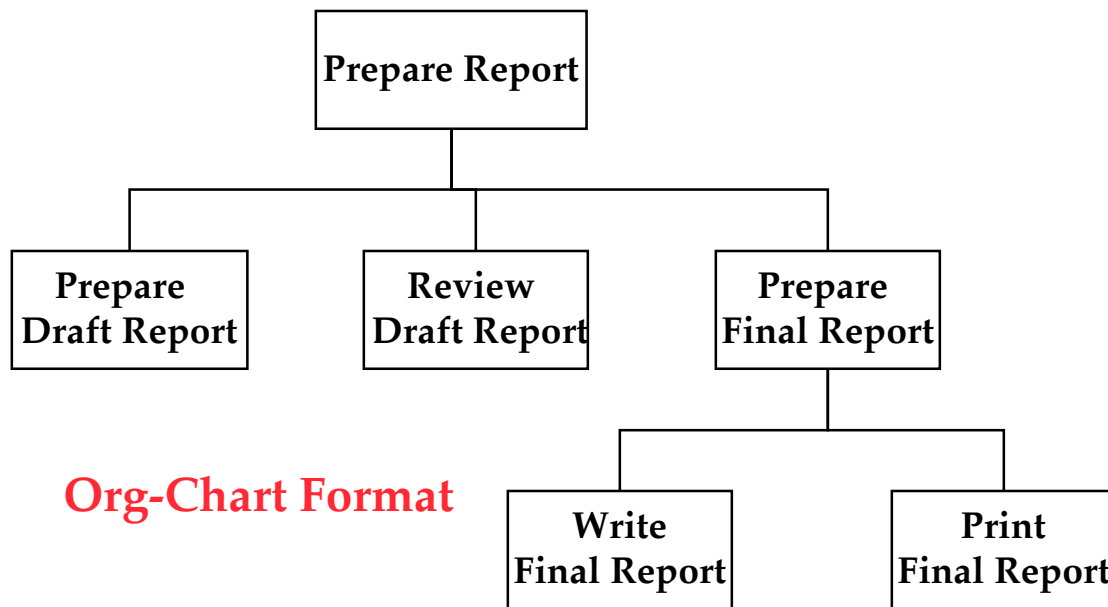
- ❖ Specify all activities required for the entire project
- ❖ Determine all tasks required to complete each activity
- ❖ If necessary, specify sub-activities required to complete each task
- ❖ Continue in this way until you have adequately broken down your project in doable tasks
- ❖ **Approach is good if**
 - ◆ You are familiar with the problem (or your team)
 - ◆ You have successfully managed a similar project in the past
 - ◆ You are not introducing new methodologies, methods or tools.

The Brainstorming WBS Development

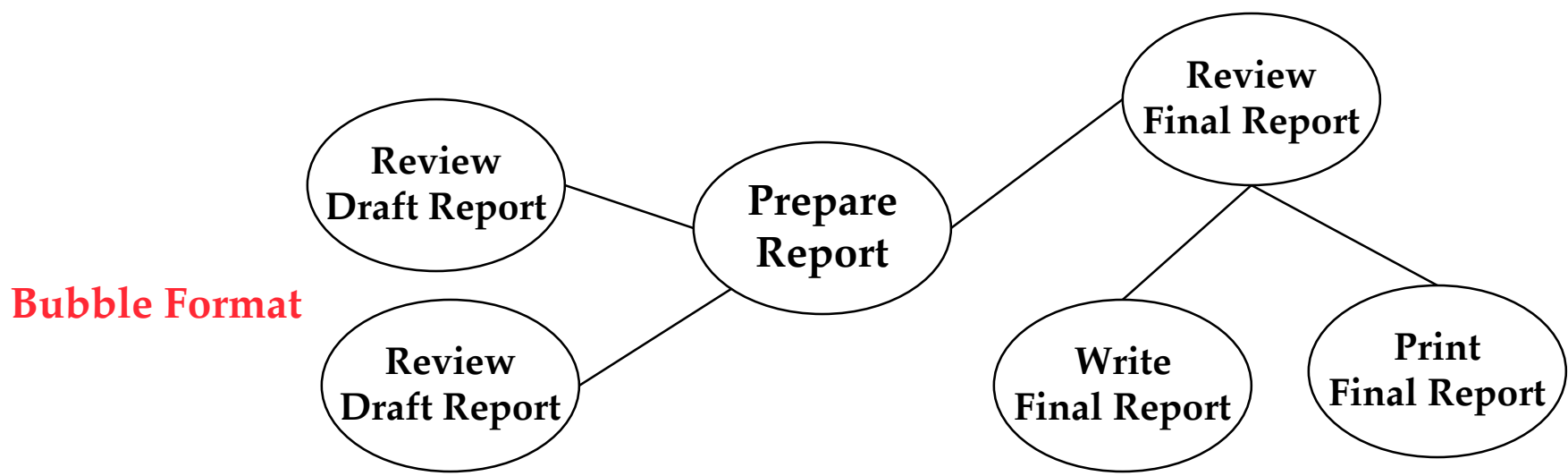
- ❖ On a single list, write any activities you think will have to be performed for your project
- ❖ “Brainstorming”:
 - ◆ Don’t worry about overlap or level of detail
 - ◆ Don’t discuss activity wordings or other details
 - ◆ Don’t make any judgements
 - ◆ Write everything down
- ❖ Then study the list and group activities into a few major categories with common characteristics
- ❖ Group identified activities into higher level activities
- ❖ Consider each category you have created and use the *top-down WBS development* to determine any additional activities you may have overlooked.

Displaying Work Breakdown Structures

- ❖ Three different formats are usually used
- ❖ **Organization-chart format**
 - ◆ Effectively portrays an overview of your project and the hierarchical relationships of different activities and tasks
- ❖ **Outline format**
 - ◆ Subactivities and tasks are indented
- ❖ **Bubble format**
 - ◆ The bubble in the center represents the project
 - ◆ Lines from the center bubble lead to activities
 - ◆ Lines from activities lead to tasks.



Outline Format



What is the best display format for WBS?

❖ **Organization-chart format:**

- ◆ Often good for a “bird view” of the project (executive summaries,...)
- ◆ Less effective for displaying large numbers of activities

❖ **Outline format:**

- ◆ Easier to read and understand if WBS contains many activities




❖ **Bubble format:**

- ◆ Effective for supporting brainstorming
- ◆ Not so good for displaying work breakdown structures to audiences who are not familiar with the project

❖ **In large projects:**

- ◆ Use bubble format to develop the WBS, then turn it into organization-chart or outline format.
- ◆ **Display activities in organization-chart format,**
- ◆ **Display sub-activities and tasks in outline format.**

Heuristics for developing high quality WBS

- ❖ Involve the people who will be doing the work in the development of the WBS
 - ◆ In particular involve the developers
- ❖ Review and include information from work breakdown structures that were developed for similar projects
 - ◆ Use a project template if possible 
- ❖ Use more than one WBS approach
 - ◆ Do project component and functional approach simultaneously
 - ◆ This allows you often to identify overlooked activities 
- ❖ Make assumptions regarding uncertain activities
 - ◆ Identify risky activities
 - ◆ These are often the activities whose times are hard to estimate. 

Choose a single WBS Approach

- ❖ *Develop* the WBS with different approaches. This is good, because it allows you to identify activities that you may overlook otherwise
- ❖ But then choose a *single* WBS approach to be used in the SPMP and for your project:
 - ◆ **Nothing confuses people more than trying to use two different work breakdown structures to describe the same project.**

How Detailed should the WBS be?

- ❖ Sometimes the activities are not clear at all, especially if
 - ◆ The requirements are not clear and/or changing
 - ◆ The software project depends on technology enablers that are promised to be available only after project kickoff
 - ◆ Hardware and software are developed in parallel (“concurrent engineering”)
- ❖ Heuristic: A project plan for 1 year - especially for an innovative software project - should not address details beyond 3 months
 - ◆ Even for the first 3 months project activities might not all be clear at all, if the requirements are unclear, change and/or the introduction of technology enablers is expected
- ❖ How should we describe a WBS for a longer project?

Doing a WBS for Long-Term Large-Scale Projects

- ❖ When developing a work breakdown structure for a long-term project (longer than 3 months), introduce at least two phases
- ❖ *Phase 1* (3 months): Plan your WBS in detail
 - ◆ **List all activities and tasks that take a week or less to complete**
- ❖ *Phase 2, Phase 3, ... (n-months)* Plan the WBS for these phases in less and less detail
 - ◆ **List activities that will take between one and two months**
- ❖ At the end of phase 1, revise phase 2 activities and plan them on the weekly level for the next 3 months
 - ◆ **Modify future activities based on the results of your first three months work**
- ❖ Continue to revise the SPMP this way throughout the project (The SPMP is an “evolving” document).




Phases in Large-Scale Projects

- ❖ Project-Initiation Phase
- ❖ Steady State Phase
 - ◆ Initial Planning phase
- ❖ Project-Termination Phase

Project-Initiation Phase: To-Do List

❖ Activities

- ◆ Meet with client, develop visionary scenario for problem statement
 - ◆ Develop initial top level design: Describe system as a set of subsystems (“subsystem decomposition”)
 - ◆ Establish staffing plan (flat staffing, ramp up)
 - ◆ Identify people: existing employees, new employees
 - ◆ Hire team members
 - ◆ Assign a subsystem to each team; establish additional cross-functional teams (e.g. architecture, documentation, demo)
 - ◆ Write problem statement (with client and other stake holders; if possible, involve project participants early)
 -  ◆ Write initial SPMP with WBS, without schedule, without budget
 - ◆ Get project plan approved
 - ◆ Kick project off with 2 documents: Problem statement and SPMP
- ❖ Duration of project-initiation-phase: Between 2-4 weeks
- ◆ **When: Before project kickoff.**

Initial Planning Phase: To-Do List

❖ Activities

- ◆ Do scouting on technology enablers that might influence the design or nonfunctional requirements
- ◆ Revise requirements and initial top level design
- ◆ Revise team structure, reassign team members
- ◆ Revise WBS and dependencies
- ◆ Establish cost and scheduling information
- ◆ Agree with client on requirements, duration and cost of the project
- ◆ Write the “project agreement” (companion document to the SPMP)

❖ Duration: About 2 weeks time

❖ When: After project kickoff, often called “planning”

- ◆ In parallel to “requirements elicitation”

Project-Termination Phase

- ❖ Do a project-review: “What went right, what went wrong”
 - ◆ also often called “project post-mortem review”
- ❖ Based on input from the post-mortem session
 - ◆ **Revise your software process, identify in particular any new activities that happened in the project**
 - ◆ **Revise your project kickoff activities**
 - ◆ **Revise the SPMP template (to be reused for your next project)**

Where are we?

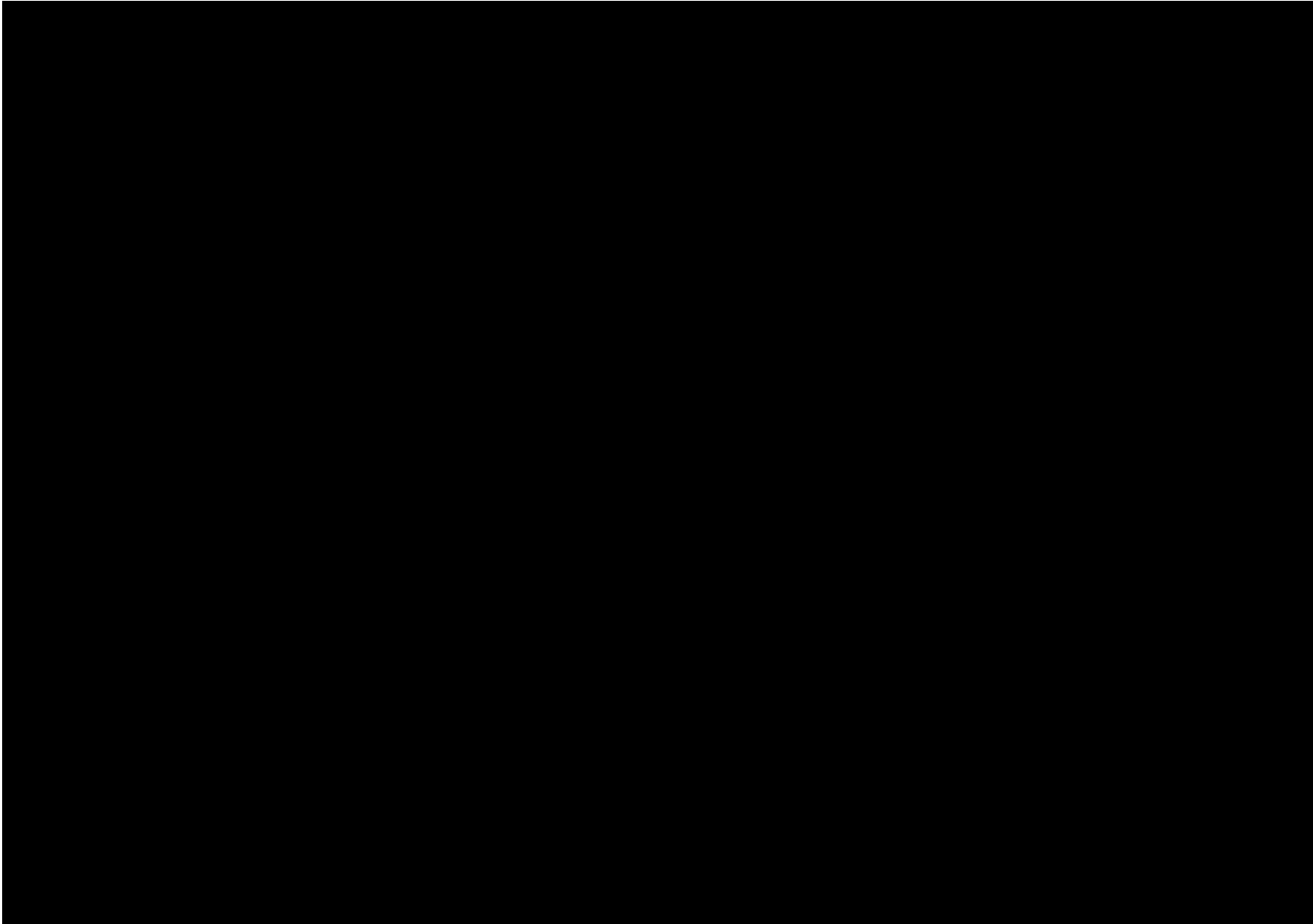
- ❖ SPMP IEEE Std 1058
- ❖ 0. Front Matter
- ❖ 1. Introduction
- ❖ 2. Project Organization
- ❖ 3. Managerial Process
- ❖ 4. Technical Process
- ❖ 5. Work Elements, Schedule, Budget
 - ✓ 5.1 Work Breakdown Structure (WBS)
 - ◆ 5.2 Dependencies between tasks
 - ◆ 5.3 Resource Requirements
 - ◆ 5.4 Budget (=> Lecture on cost estimation)
 - ◆ 5.5 Schedule
- ❖ Optional Inclusions

Readings

- ❖ Literature used for this lecture
 - ◆ [IEEE Std 1058] Standard for Software Project Management Plans
 - ◆ Stanley E Portny, *Project Management for Dummies, Hungry Minds, 2001, ISBN 0-7645-5283-X*
 - ◆ [Bruegge-Dutoit 2000], Chapter 11 Project Management

Summary

- ❖ Different approaches to develop a WBS
 - ◆ **Product Approach**
 - ◆ **Functional Approach**
 - ◆ **Geographical Approach**
 - ◆ **Organizational Approach**
- ❖ Top down and bottom up WBS development
- ❖ Heuristics for developing good WBS
- ❖ WBS for Large Projects



Heuristic: Use Templates

- ❖ Try to derive the SPMP from a template
 - ◆ A template reflects the cumulative experience gained from doing numerous projects of a particular type
 - ◆ Using templates can save you time and improve your accuracy
- ❖ When developing templates, develop them for frequently performed tasks (reviews, meetings, ...)
- ❖ Develop “Checklists”:
 - ◆ Develop and modify your WBS templates from previous projects that worked, not from plans that looked good
 - ◆ Use templates as starting points, not as ending points
 - ◆ Continually update your templates to reflect the experience gained from performing different projects.



Heuristic: Develop always more than one WBS

- ❖ Consider to create more several different hierarchies with different categories for your work breakdown structure.
 - ◆ **Having two or more different perspectives helps you identify activities you may overlook**
- ❖ Good starting point are the following hierarchies:
 - ◆ **Entity-oriented decomposition**
 - ◆ **Activity-oriented decomposition**
- ❖ Example: You are running your first object-oriented project.
 - ◆ **Develop a WBS based on the project documents**
 - ◆ **Develop a WBS based on the software process activities.**

Heuristic: Identifying Risky activities

- ❖ When you identify activities for a work breakdown structure, you can also identify the risks in your project
- ❖ Risks are usually associated with “unknown information”
- ❖ Unknown information comes in two flavors
 - ◆ A “**known unknown**”: Information that you don’t have but someone else does
 - ◆ Find out who has the information and determine what the information is. (Interviews, phone calls, tasks analysis)
 - ◆ An “**unknown unknown**”: Information that you don’t have because it does not yet exist
 - ◆ Develop **contingency plans** for each of these risks
 - ◆ These contingency plans need be followed when you find out the information does not exist
- ❖ Describe these risks in SPMP 3.3 Risk Management.



Risk Management Examples

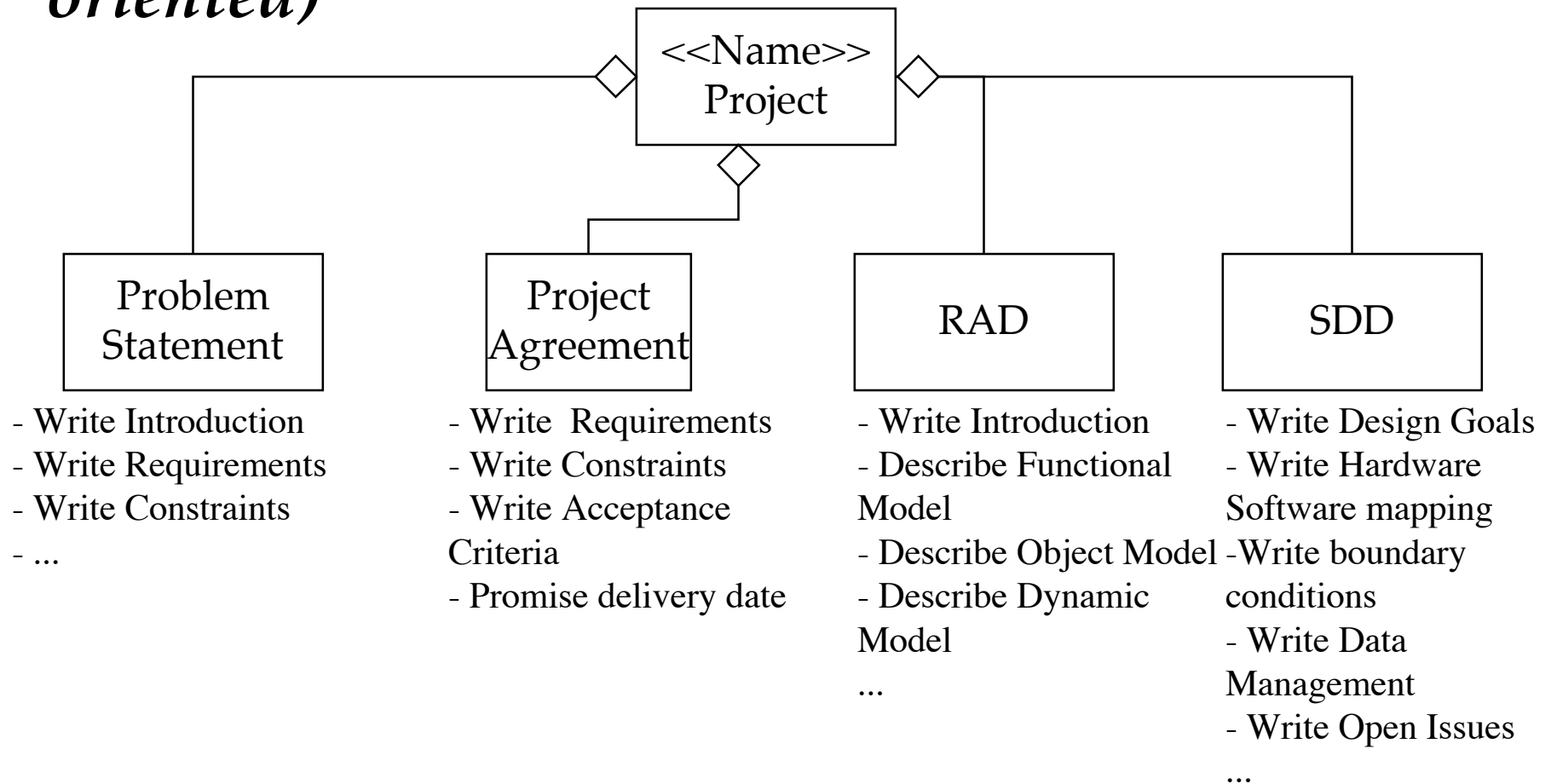
- ❖ Risk: Members in key roles leave the project.
 - ◆ *Contingency Plan?*
 - ◆ Roles are assigned to somebody else. Functionality of the system is renegotiated with the client.
- ❖ Risk: The project is falling behind schedule.
 - ◆ *Contingency Plan?*
 - ◆ Extra project meetings are scheduled.
- ❖ Risk: Team 1 cannot provide functions needed by team 2.
 - ◆ *Contingency Plan?*
 - ◆ **A: We drop the functionality.**
 - ◆ **B: The liaisons of both teams get together to solve this problem**
- ❖ Risk: The planned PDA will not be available.
 - ◆ *Contingency Plan?*
 - ◆ **We will use an IPAQ instead.**

Risk Management Examples ctd

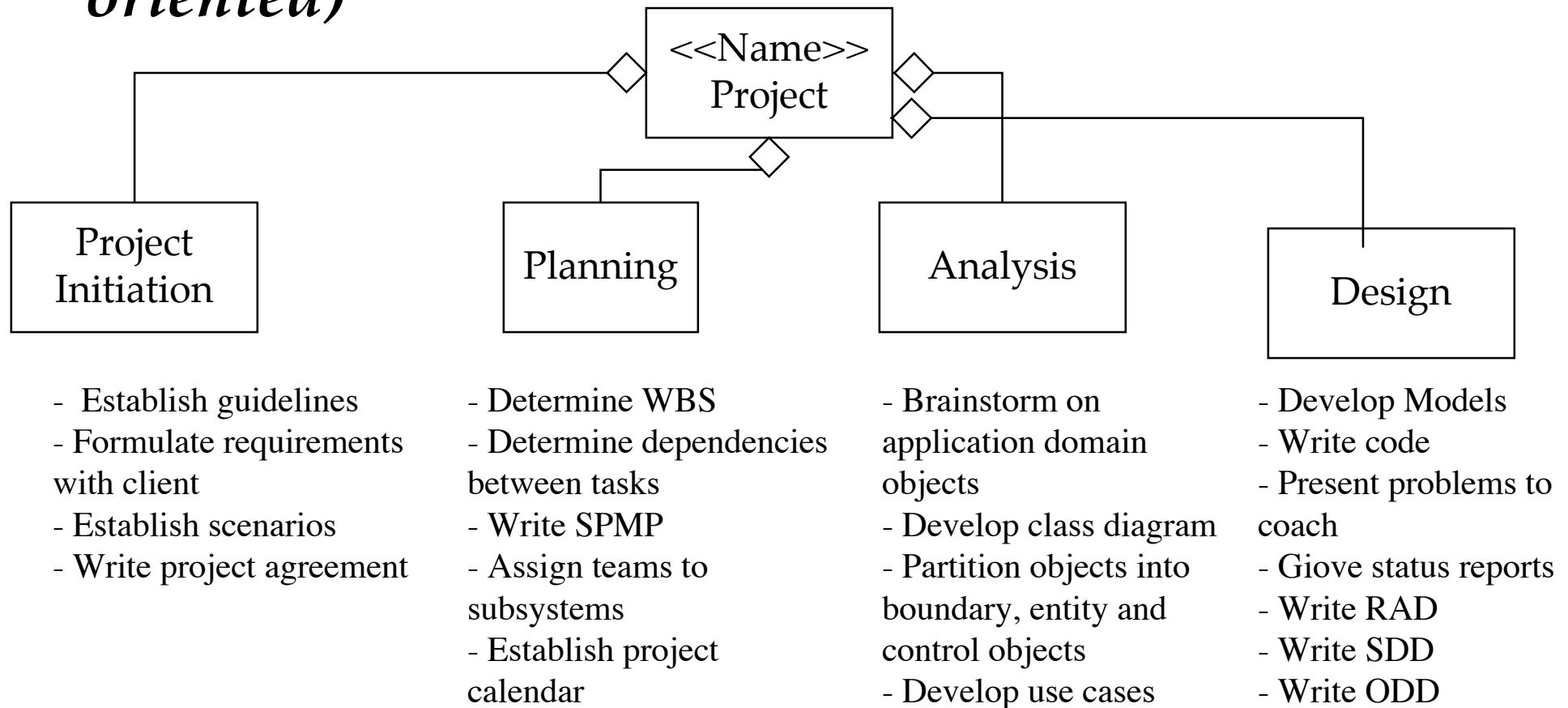
- ❖ Risk: The selection of the database system takes too much time
 - ◆ *Contingency Plan?*
 - ◆ **The Database team uses a bridge pattern and provides a test stub to be used by the other teams for data access while the selection process goes on.**
- ❖ Risk: The customer is not available for discussing and reviewing the user interface during development.
 - ◆ *Contingency Plan?*
 - ◆ **Make the design decisions that we feel are appropriate**
- ❖ Risk: No suitable wireless library can be found.
 - ◆ *Contingency Plan?*
 - ◆ **The wireless team develops its own library**



WBS Based on Project Documents (Entity-oriented)



WBS Based on Software Process (Activity-oriented)



Question: Which activities mentioned in the WBS based on Project documents is left out in the WBS based on Software Process?

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 Person Months (PM)
 - ◆ **Medium project (300 person-month):** at least 1% or 3 PMs
 - ◆ **Large project (7000 person-month):** at least 0.2 % or 15 PMs

Source: Barry Boehm, Software Economics (rather out-of-date)

Backup Slides

