

TUM

Project Organization

30 May 2006

Bernd Bruegge

Technische Universität München

Institut für Informatik

Lehrstuhl für Angewandte Softwaretechnik

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

Where are we?

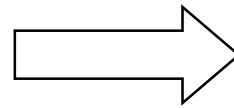
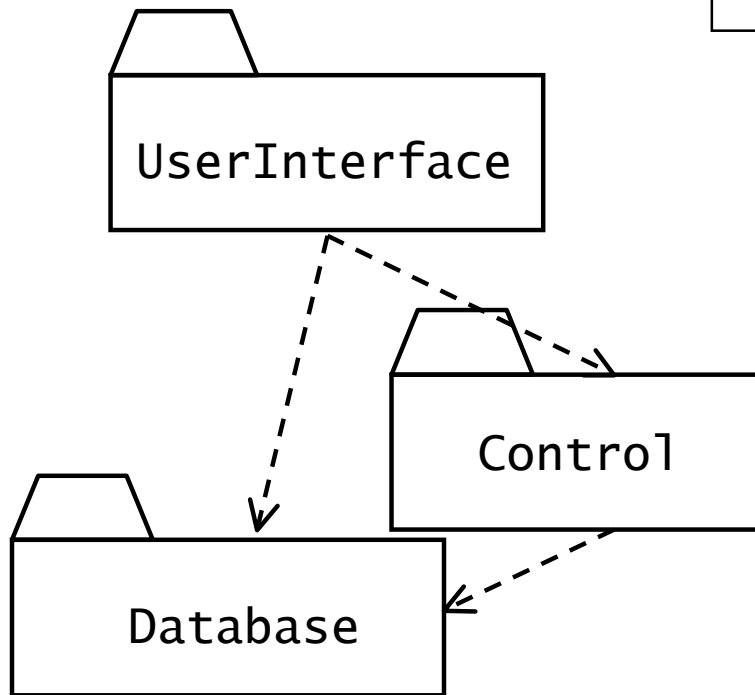
- Software Process
 - Configuration Management
 - System Testing
 - Software Lifecycle Modeling
 - Rationale Management
 - Methodologies
- Software Project Management
 - Work Breakdown Strategies
 - Estimation
 - Organization
 - Scheduling
 - Planning and Controlling

Where are we?

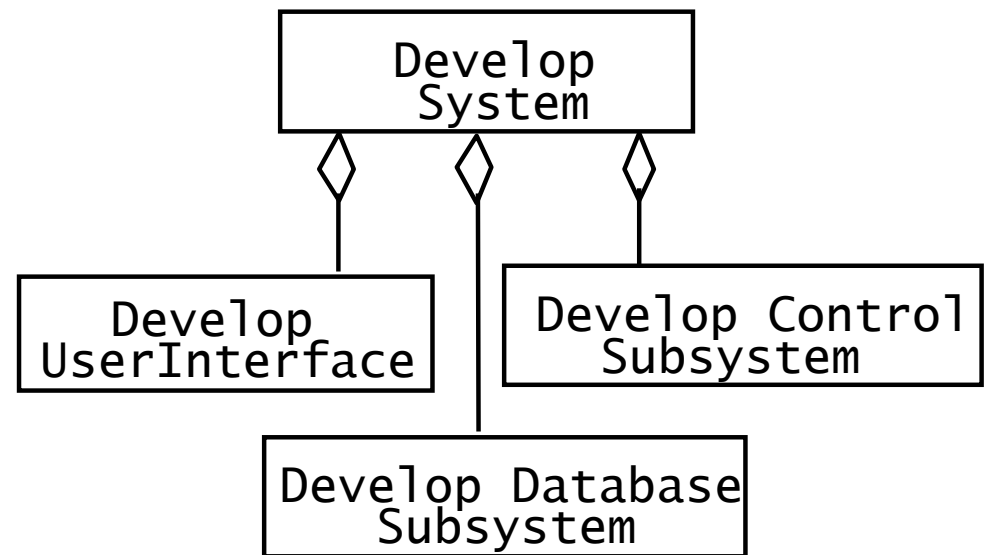
- Software Process
 - ✓ Configuration Management
 - System Testing
 - Software Lifecycle Modeling
 - Rationale Management
 - Methodologies
- Software Project Management
 - ✓ Work Breakdown Strategies
 - ✓ Estimation
 - **Organization**
 - Scheduling
 - Planning and Controlling

Setting up a Project

Establish Subsystem decomposition (“Top-Level Design”)

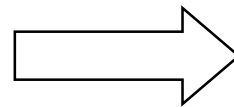
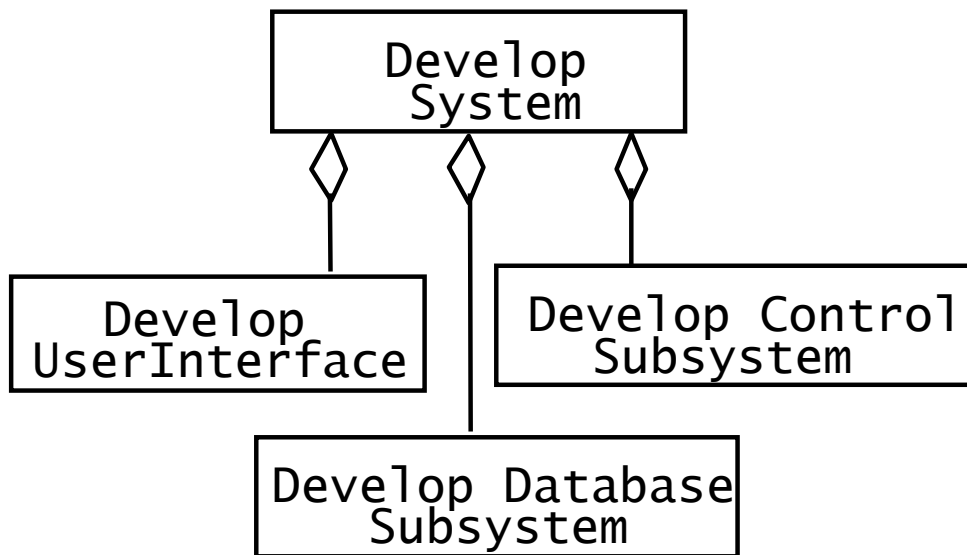


Determine Work Breakdown Structure

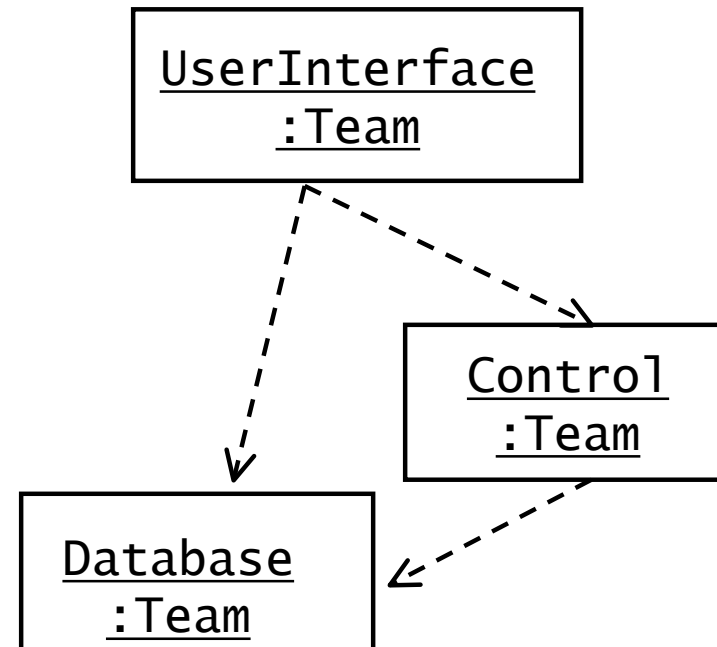


Project Organization with 3 Teams

Work Breakdown Structure



Set up Teams



Three Steps

1. Determine subsystem decomposition
 - Software Architecture, top-level design
2. Determine work breakdown structure
3. Establish teams

Establishing a Project-based Organization

- Every time, you set up a project, the same set of organizational issues appear
 - How should the teams be organized?
 - Who are the key players?
 - What roles and responsibilities do they assume?
 - Who is in charge?
 - What is the information flow between roles?
 - What are the benefits and challenges (“pros and cons”)?
- **Architecture-centric project management**
 - Formulate software architecture (documented in the system design document) *simultaneously* with project organization (documented in the SPMP)
 - Good Book: [Paulish, 2001].

Group vs. Team

- **Group**
 - A set of people who are assigned to a *common task* and who work *individually* to accomplish their assignment
- **Team**
 - A small group of people working on the *same problem or subproblem* in a project. The team members - also called participants - *depend on one another* to do their tasks
- **Project Team**
 - Based on the premise that every participant must make a valuable contribution to the project.

Organization

- **Organization**
 - A set of *organizational units* and their different *relationships* with each other
- Organizational units can be organized according to many different categories
 - by function, by project type, ...
- Typical examples of organizational units:
 - Functional organization
 - Research, Development, Marketing, Sales
 - Project-based organization
 - Project 1, Project 2, Project 3.

Structures in Organizations

- An organization usually has 3 different types of associations between organizational units
 - **Reporting structure**
 - Shows how status information is reported
 - **Decision structure**
 - Shows how decisions are propagated
 - **Communication structure**
 - Shows how information is exchanged.

Roadmap for the Lecture

- Discussion of different organization forms
 - Functional organization
 - Project-based organization
 - Matrix organization
- People and roles in organizations
 - Project manager, team member, developer, analyst, ...
 - Binding roles to people
 - Responsibility, Authority, Accountability and Delegation
- Relationships between roles
 - Hierarchical and nonhierarchical organizations
- Identifying people
 - Audience list, drivers, supporters, observers
 - Involvement of audience list members during the lifetime of a project.

Functional Organization

- In a functional organization people are grouped into **departments**, each of which **addresses an activity** (“function”)
- Examples of departments
 - ▶ Traditional companies: Finance, production, sales, marketing
 - Software companies: Analysis, design, integration, testing
- Properties of functional organizations
 - Projects are pipelined through the departments.
 - Example: The project starts in research, moves to development, then moves to production
 - Different departments often address identical needs
 - Example: Configuration management, IT infrastructure
 - Only few participants are involved in the complete project.

Properties of Functional Organizations

- **Advantages:**
 - Members of a department have a good understanding of the functional area they support
- **Disadvantages:**
 - It is difficult to make major investments in equipment and facilities
 - High chance for overlap or duplication of work among departments.

Project-based Organization

- In a **project-based organization** people are assigned to projects, each of which has a problem to be solved within time and budget
- Key properties of project-based organizations
 - Teams are assembled for a project as it is created
 - Each project has a project leader
 - All participants are involved in the complete project
 - Teams are disassembled when the project terminates.

Properties of Project-based Organizations

- **Advantages**

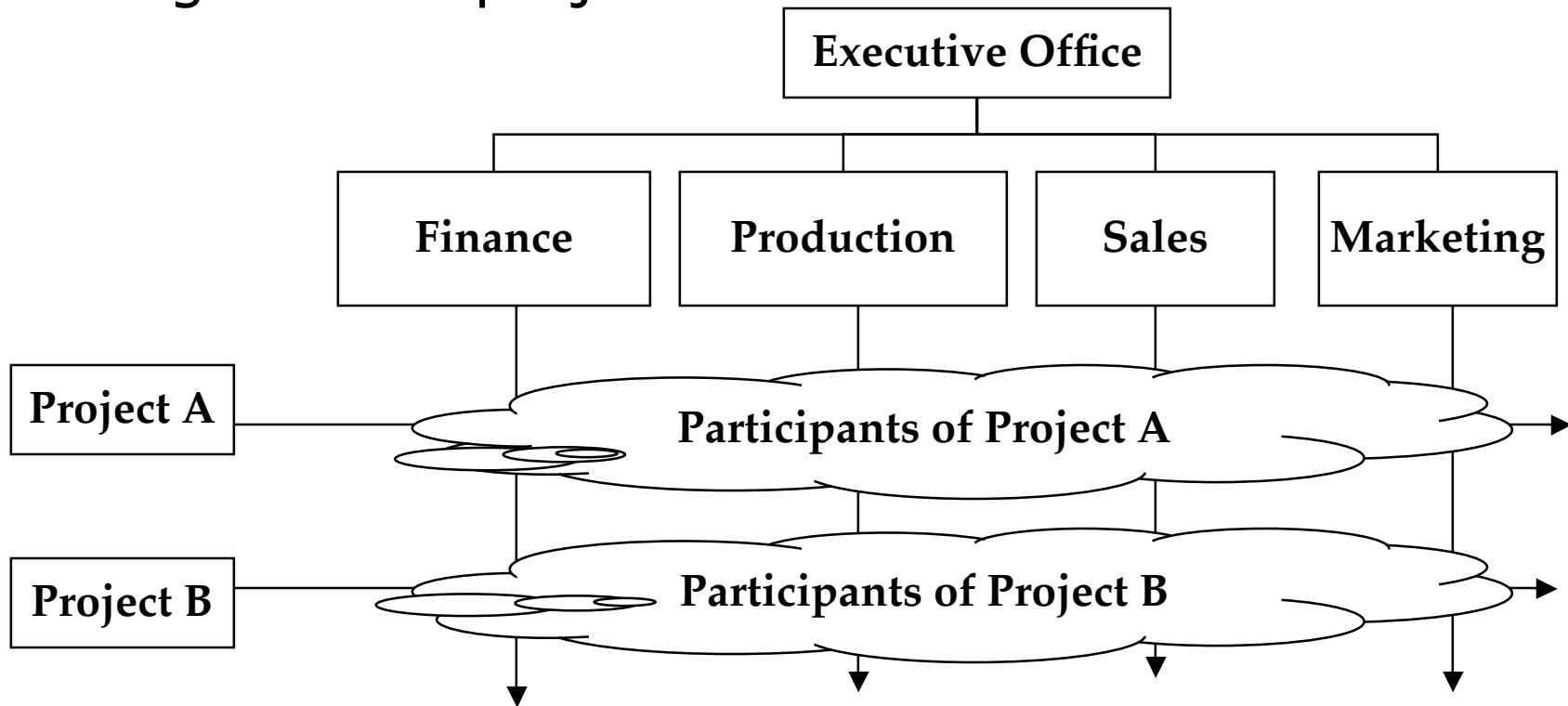
- Very responsive to new requirements (because the project is newly established and can be tailored around the problem)
- New people can be hired who are familiar with the problem or who have special capabilities
- There is no waste of staff workload

- **Disadvantages**

- Teams cannot be assembled rapidly. Often it is difficult to manage the staffing/hiring process
- Because there are „no predefined lines“, roles and responsibilities need to be defined at the beginning of the project.

Matrix Organization

- In a **matrix organization**, people from different departments of a functional organization are assigned to work on one or more projects
- Project manager and participants are usually assigned to a project < 100 % of their time.



Properties of Matrix Organizations

- **Advantages**
 - Teams for projects can be assembled rapidly
 - Rare 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 usually are not familiar with each other
 - Team member have different working styles
 - Team members must get used to each other.

Challenges in Matrix Organizations

- Team members working on multiple projects have competing demands for their time
- Team members must respond to two different bosses with different focus:
 - Focus of the functional manager:
 - Assignments to different projects, performance appraisal
 - Focus of the project manager:
 - Work assignments to project members, support of the project team
- Multiple work procedures and reporting systems are used by different team members.

When to use a Functional Organization

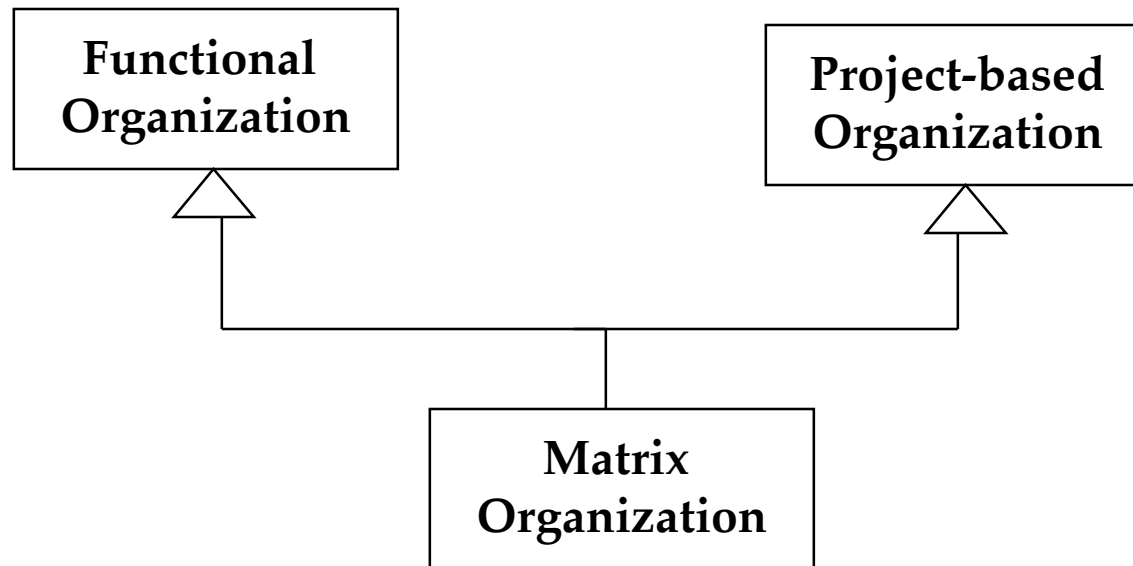
- Projects with high degree of certainty, stability, uniformity and repetition
 - Requires little communication
 - Role definitions are clear

The more people on a project, the more the need for a formal structure.

When to use a Project-based Organization

- Project has high degree of uncertainty
 - Open communication needed among members
 - Roles are defined on project basis
- When?
 - Requirements change during development
 - New technology appears during project.

Meta-Model for Organizations



Roadmap for the Lecture

✓ We discussed different organization forms

- Functional organization
- Project-based organization
- Matrix organization

→ Now we will talk about the different roles played by people in these organizations

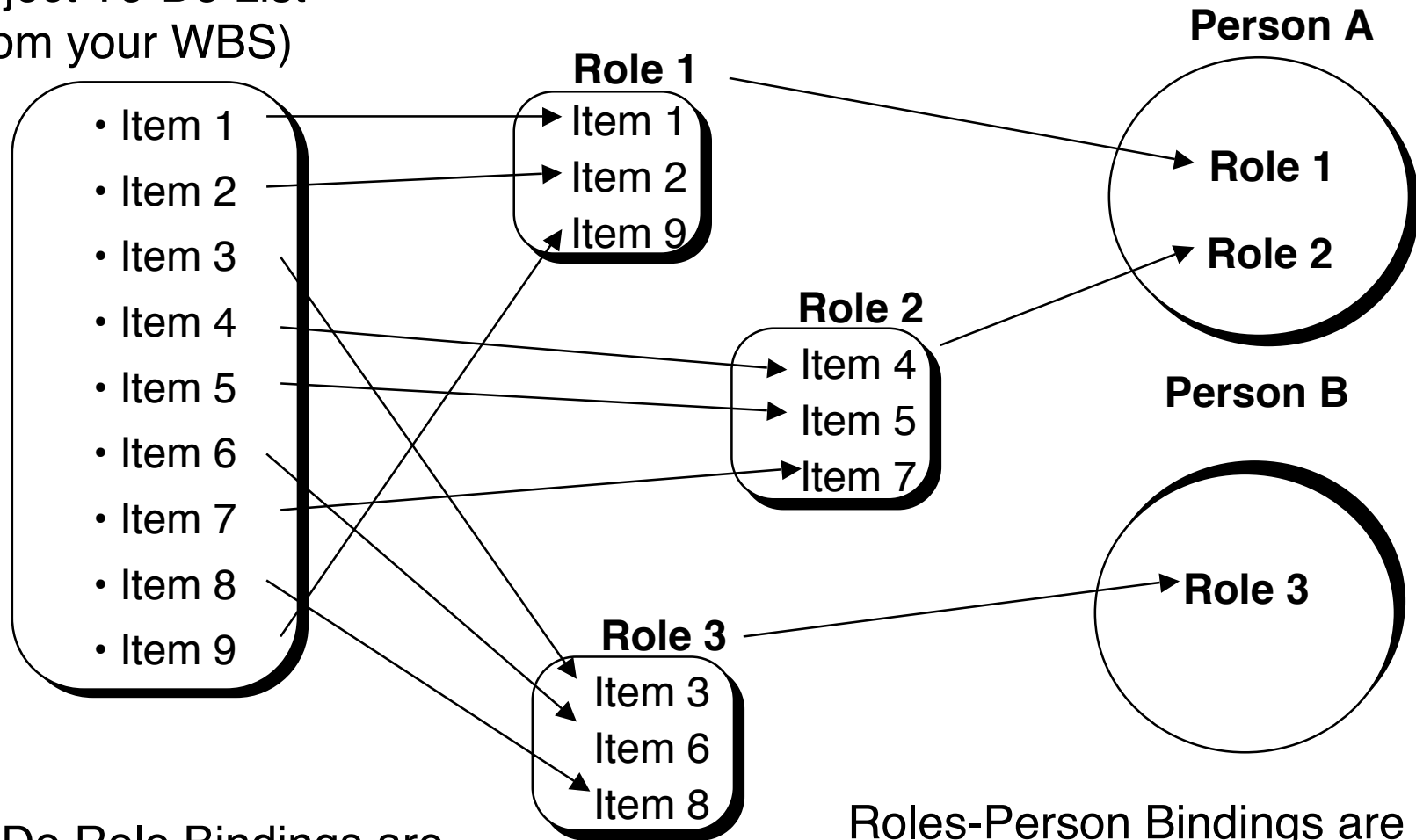
- Project manager, team member, developer, analyst,
- Responsibility, Authority, Accountability and Delegation.

Definition Role

- A **role** is a set of commitments to achieve specific results
- A role is instantiated during a project and assigned to one or more participants
- Instances of roles are often also called **players** („who are the key players?“) or **stakeholders**.

Binding Roles To People

Project To-Do List
(from your WBS)



To-Do Role Bindings are made during Project-Initiation Phase

Roles-Person Bindings are made during Initial Planning phase (First team meeting, etc ...)

Flexibility of Organizations

- An organization is **flexible**, if it allows “late” or even “dynamic” bindings
 - of roles to people and
 - of the information flow between roles
- **Late binding**
 - Organizational units and information flows are established just in time for the project
 - Cannot be changed after project kickoff
- **Dynamic binding**
 - The organizational relationships change over time
 - |They can be changed anytime.

Binding Roles to People: Multiplicities

- One-to-One
 - Ideal but often not worth to be called a project
- Many-to-Few
 - Each project member assumes several roles ("hats")
 - Danger of over-commitment
 - Need for load balancing
- Many-to-"Too-Many"
 - Some people don't have significant roles
 - Bystanders
 - People loose the touch with project.

Binding Roles to People: Main abstractions

- **Responsibility**
 - The commitment to achieve specific results
 - Redefinition of role: A **role** is a set responsibilities
- **Authority**
 - The ability to make the binding decisions between roles and people
- **Accountability**
 - Tracking a task performance to a person
- **Delegation**
 - Binding a responsibility assigned to one person (including yourself) to another person
- Question: If you model these abstractions, which modeling concepts would you use?

Three Reasons for Delegation

- **Time Management:** Free yourself up to do other tasks
- **Expertise:** Select a better qualified person to make the decision
- **Training:** To develop another person's ability to handle additional assignments.

Authority vs. Responsibility

- Both are agreements
 - Before you start a project, you agree on
 - who can make decisions and
 - who will ensure that particular results are achieved
- Difference:
 - **Authority is activity-oriented:** It focuses on *process* such as activities and tasks
 - **Responsibility is entity-oriented:** It focuses on *outcome* such as work products and deliverables.


Responsibility vs. Accountability

- Both are entity-oriented (focus on the result!):
- Difference:
 - Responsibility is an agreement done before a task started
 - Accountability is investigated after a task is performed
- A person who is responsible is also accountable
- A person who is not responsible is not accountable
 - **Scapegoating**: Making somebody accountable who was not responsible
- Delegation of responsibility is associated with risks.

Risks when Delegating Responsibility

- **Risk:** *Responsible person is over-committed*
 - Project Management Issues:
 - Person does not have enough time to handle all roles
 - Person is making too many key decisions
 - What if this person leaves during the project?
- **Risk:** *The project manager has no longer any responsibilities ("everything was delegated")*
 - Will the project manager understand the status reports?
- **Risk:** *The outcome requires additional approvals*
 - Does anyone else have to approve the outcome?
 - Are there too many people involved in the approvals?
 - The estimated duration of the activity may be too optimistic, because it is overlooked, that the approval involves many people.

Key Roles in Projects

- **Project Manager:** The person responsible for the successful completion of the project 
- **Team Member:** Participants responsible for performing activities and tasks (in a project or matrix organization)
- **Functional Manager:** The team member's supervisor in the department (in a functional organization)
- **Upper management:** People in charge of the departments or projects ("program manager")

In the following we focus only on roles in project-based organizations.

General Responsibilities of Team Members

- Technical responsibilities:
 - Perform assigned tasks within time
 - Acquire technical skills and knowledge needed to perform the work
- Managerial responsibilities
 - Identify situations and problems that might affect the tasks
 - Keep others informed about your progress and problems you encounter.


Typical Team Member Roles

- Project Management
 - Coach
 - Team leader
 - API Liaison
 - Planner
- Meeting Management
 - Minute Taker
 - Scribe
 - Primary facilitator
- Development
 - Analyst
 - Designer (Software Architect)
 - Programmer
 - Tester
 - Maintainer
 - Trainer
 - Document Editor
 - Web Master
 - Configuration Manager

A Taxonomy for Project Roles

- **Management role**
 - Organization and execution of the project within constraints. Examples: project manager, team leader
- **Development role**
 - Specification, design and construction of subsystems. Examples: Analyst, software architect, programmer
- **Cross functional role**
 - Execute project functions. Examples: API Liasion, configuration manager
- **Consultant role**
 - Supports in areas where project participants lack expertise. Examples: End user, client, application domain specialist (problem domain), technical consultant (solution domain)
- **Promoter role**
 - Deals with change in organization, application/solution domain, software process.

Promoter

- **Promoters** are self appointed individuals who identify themselves with the outcome of the project
 - They are member of the corporate organization and may not necessarily be directly involved with the project
 - Instead, they are the interface to the rest of the corporate organization
- Able to push specific changes through the existing organization which are needed to make the project a success
- Power promoter, knowledge promoter, process promoter. 

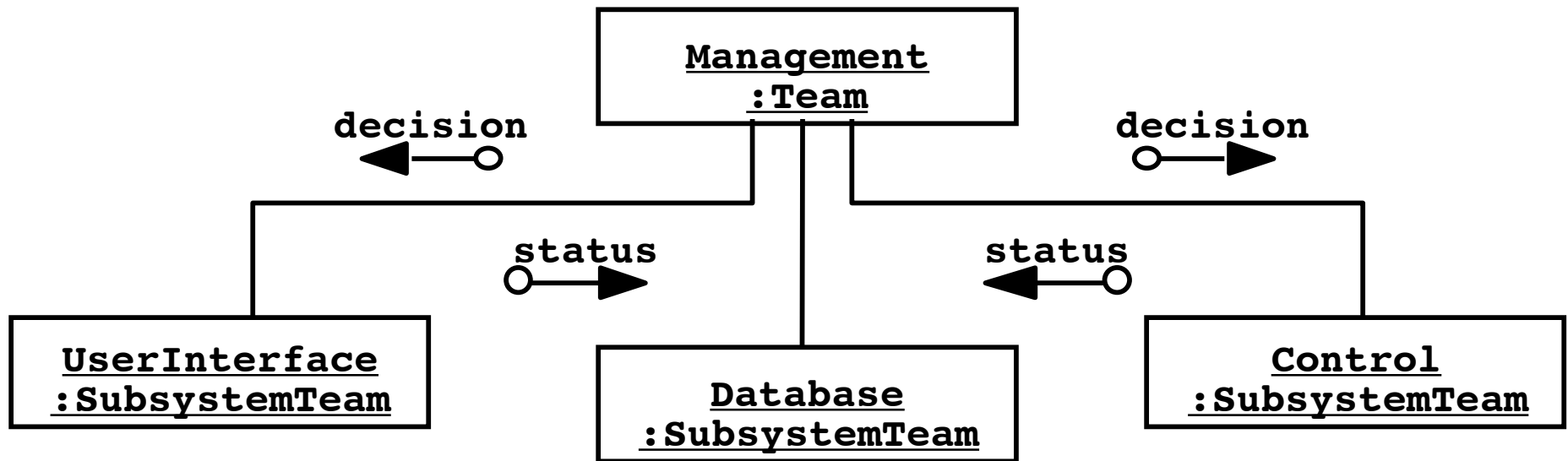
Roadmap for the Lecture

- ✓ We first discussed different organization forms
 - Functional Organization
 - Project Organization
 - Matrix Organization
- ✓ Then we talked about the different roles played by people in these organizations
 - “Dynamic model” of roles: Responsibility, Authority, Accountability and Delegation
 - Taxonomy of roles: Project Manager, Team Member, Upper Management,.....,Promoters
- ➔ Now we discuss different types of **relationships between the roles**
 - Hierarchical Organizations
 - Nonhierarchical Organizations.

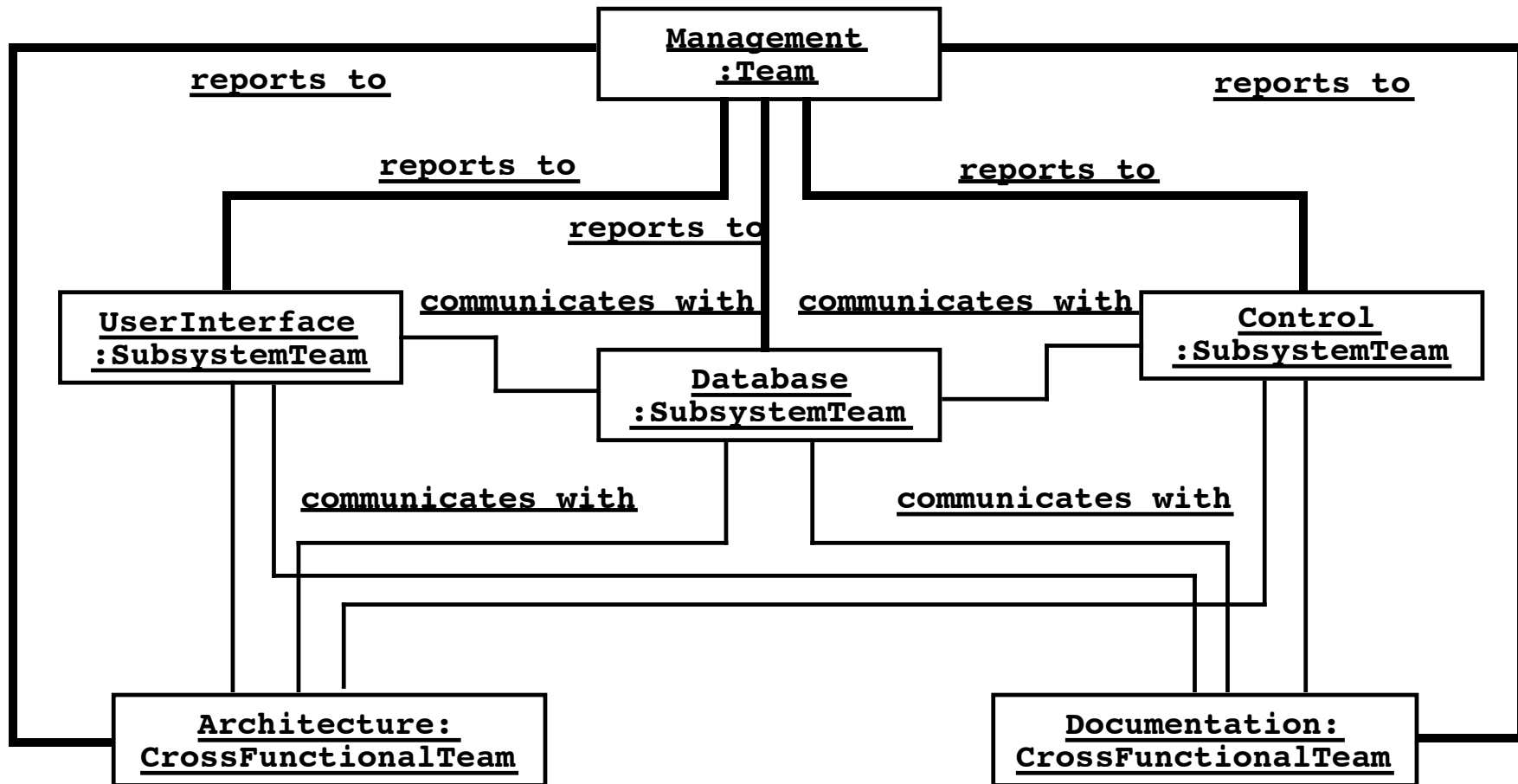
Relationships between Roles

- Organizations can have many different types of associations between roles
- The three most important associations for project organizations are: *Reporting, decision making and communicating*
- **Reporting association**
 - Used for reporting status information
- **Decision association**
 - Used for propagating decisions
- **Communication association**
 - Used for exchanging information needed for decisions (e.g., requirements, design models, issues).


An Organization with Reporting and Decision Structure



An Organization with Distinct Reporting, Decision and Communication Structures



Hierarchical Organization

- Often also called *centralized organization*.
Examples: Military, church, traditional businesses 
- Key properties
 - The organization has a tree structure
 - Decisions are made at the root and communicated to the leaf nodes
 - The decision association is also used for reporting and communication.

Advantages of Hierarchical Organizations

- Centralized control over project selection
- One set of management and reporting procedures for all project participants across all projects
- Established working relationships among people
- Clearly established lines of authority to set priorities and resolved conflicts
- Authority to pressure people to honor their action items
- Clearly defined career path.

Disadvantages of Hierarchical Organizations

- **Slow response time**
 - Evaluating and approving change requests takes too long because of long reporting/decision lines
- **Difficult to manage the workload of the people**
 - People are fulltime members of the organization, but projects don't come in a steady stream
 - Project might not require the available people
- **Problems with application or solution domain**
 - People are hired for their technical proficiency in a specialty that the organization normally performs.
 - Often they have only limited experience, if the problem to be solved is outside their field of expertise.

Nonhierarchical Organizations

- An organization that can be described with a general graph structure
 - different edges for the decision, reporting and communication flows
 - Decisions can be made at various nodes in the graph.



Observations on Organizational Structures

- **Hierarchical structure**
 - “Reports”, “Decides” and “Communicates-With” are all mapped onto the same association
 - Does not work well with iterative and incremental software development processes
 - Manager is not necessarily always right
- **Nonhierarchical structure**
 - “Reports”, “Decides” and “Communicates-With” are modeled as different associations
 - Cuts down on bureaucracy
 - Reduces development time
 - Decisions are expected to be made at each level
 - Hard to manage.

Final Topic for Today: Identifying People

✓ Organizational Structures

- Functional, Project and Matrix Organizations

✓ Taxonomy for roles (Object model)

- Project Manager, Team members, upper management, ...

✓ States of a role (Dynamic model)

- ✓ Responsibility, Authority. Accountability and Delegation

✓ Project functions involving roles (Functional model)

- Decision making, status reporting, communication

»»»» Another taxonomy of people

- Audience List, Drivers, Supporters, Observers
- Involvement of audience members during the lifetime of a project.

Identifying People

- **Audience List:** A list of people or groups of people that support the project or are simply interested in it
- As soon as you start thinking about a project, you should start the audience list
- It is a good idea to start with a template
 - Audience List Template.

Categories for an Audience List Template

- Internal
 - Project manager
 - Upper management
 - Requester
 - Team members
 - People with special knowledge
- External
 - Clients or customers
 - Collaborators
 - Vendors, suppliers and contractors
 - Regulators
 - The general public
- Support Groups
 - Human Resources
 - Legal services
 - Contracting
 - Finances
 - Security
 - Computing Facilities
- End users of the project's deliverables
- People who will maintain or support the deliverables.

Guidelines for the Audience List

- Use a template that worked well in a previous project
- Speak with a wide range of people
- Encourage project participants to identify additional candidates
- Instantiate instances from each category with position and name
- Separately include a person's name for every different role played by him or her
- Allow sufficient time to developing the audience list (mainly during project initiation time).

Other Categories for the Audience List

- **Drivers**
 - People who have some say in defining the results of the project
- **Supporters**
 - People who help to perform the activities and tasks of the project
- **Observers**
 - People who are interested in the activities and results of the project
- **Project Champion**
 - A person who strongly supports the project, even advocates it in disputes
 - Takes whatever is necessary to help ensure the successful completion of the project.

Methods to keep the Audience involved

- **One-on-one meetings**
 - Formal and informal meetings with one or two other participants about project issues
- **Group meetings**
 - Planned session for some or all project team members (weekly meeting), the client (reviews) or other members of the audience list
- **Informal written correspondence**
 - Notes, memos, letters and e-mail to document informal discussions and to share important project information
- **Written approvals**
 - Formal written agreements about a work product, schedule, resource commitment or a technical approach.

Summary

- Organization: A graph with nodes (organizational units) and different type edges (information structures)
 - Functional, project-based and matrix organization
- Teams are the key to project-based organizations
- Flexibility of organizations
 - Dynamic binding of responsibilities to people
- Project roles in project organizations
- Authority, Responsibility, Accountability, Delegation („dynamic model of the organization“)
 - Delegation involves risks.

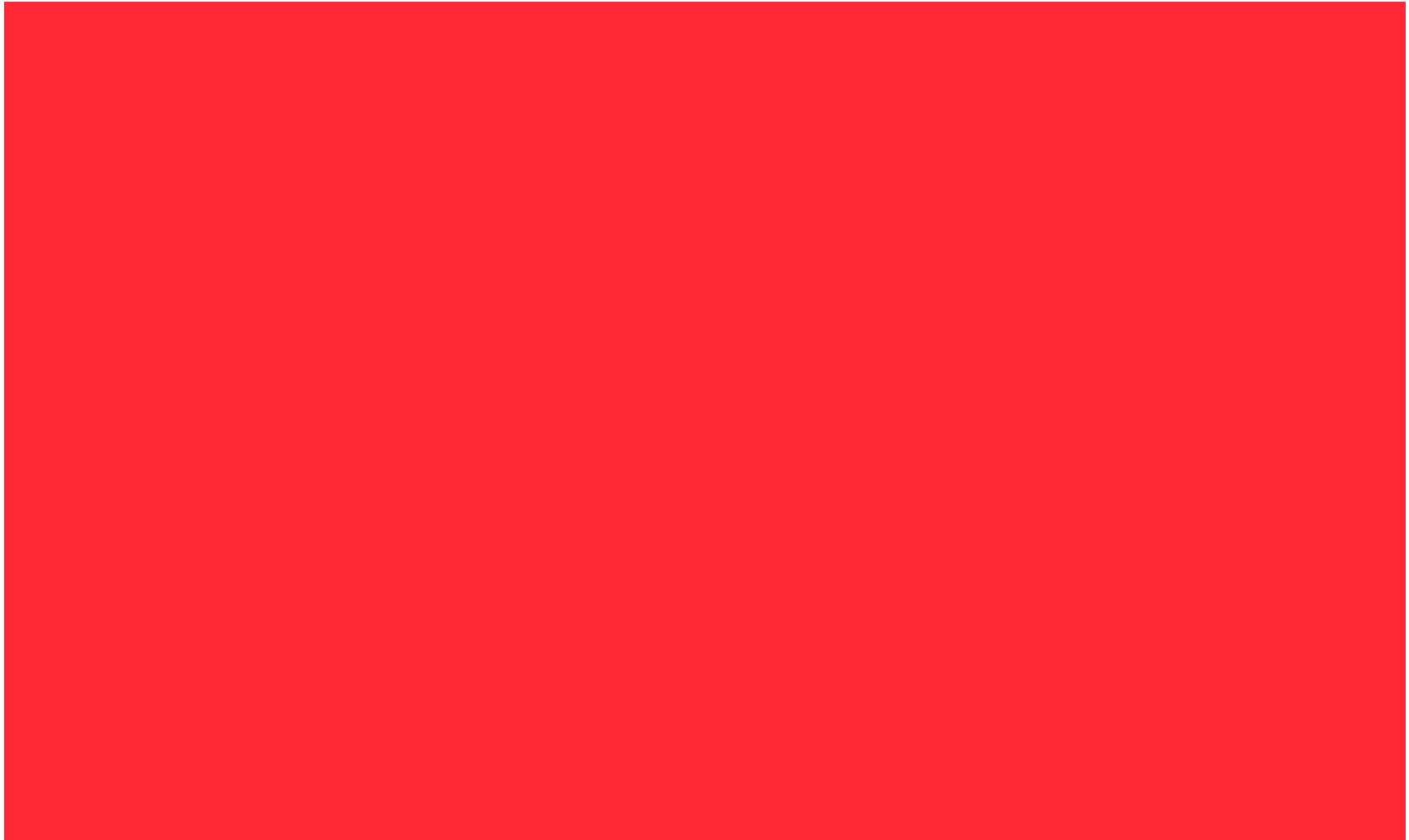
References

- Literature used for this lecture
 - [Bruegge-Dutoit 2003], Ch. 11 Project Management
- Additional readings:
 - [Paulish, 2001] D. J. Paulish, Architecture-centric Software Project Management , SEI Series in Software Engineering, Addison-Wesley, 2001
 - [Raymond, 1998] E. Raymond, The cathedral and the bazaar, <http://www.tuxedo.org/~esr/writings/cathedral-bazaar/cathedral-bazaar.html>, 1998
 - [Brooks, 1995] F. P. Brooks, The Mythical Man Month: Anniversary Edition: Essays on Software Engineering. Addison-Wesley, Reading, MA, 1995
 - [Weinberg, 1971] G. M. Weinberg, The Psychology of Computer Programming, Van Nostrand, New York, 1971.

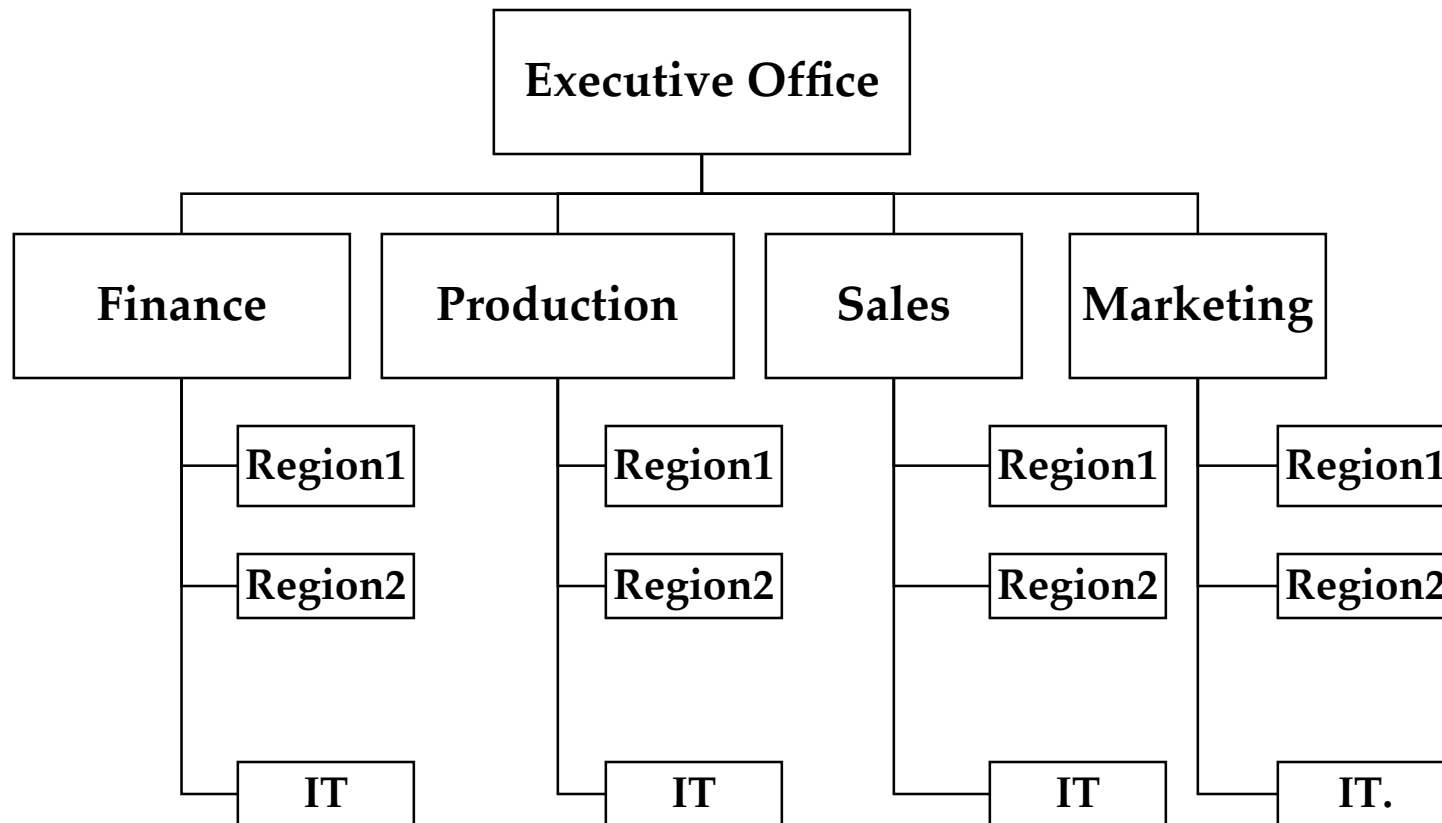
Tomorrow's Exercise

- Handout available today
- Please pick it up.

Backup slides

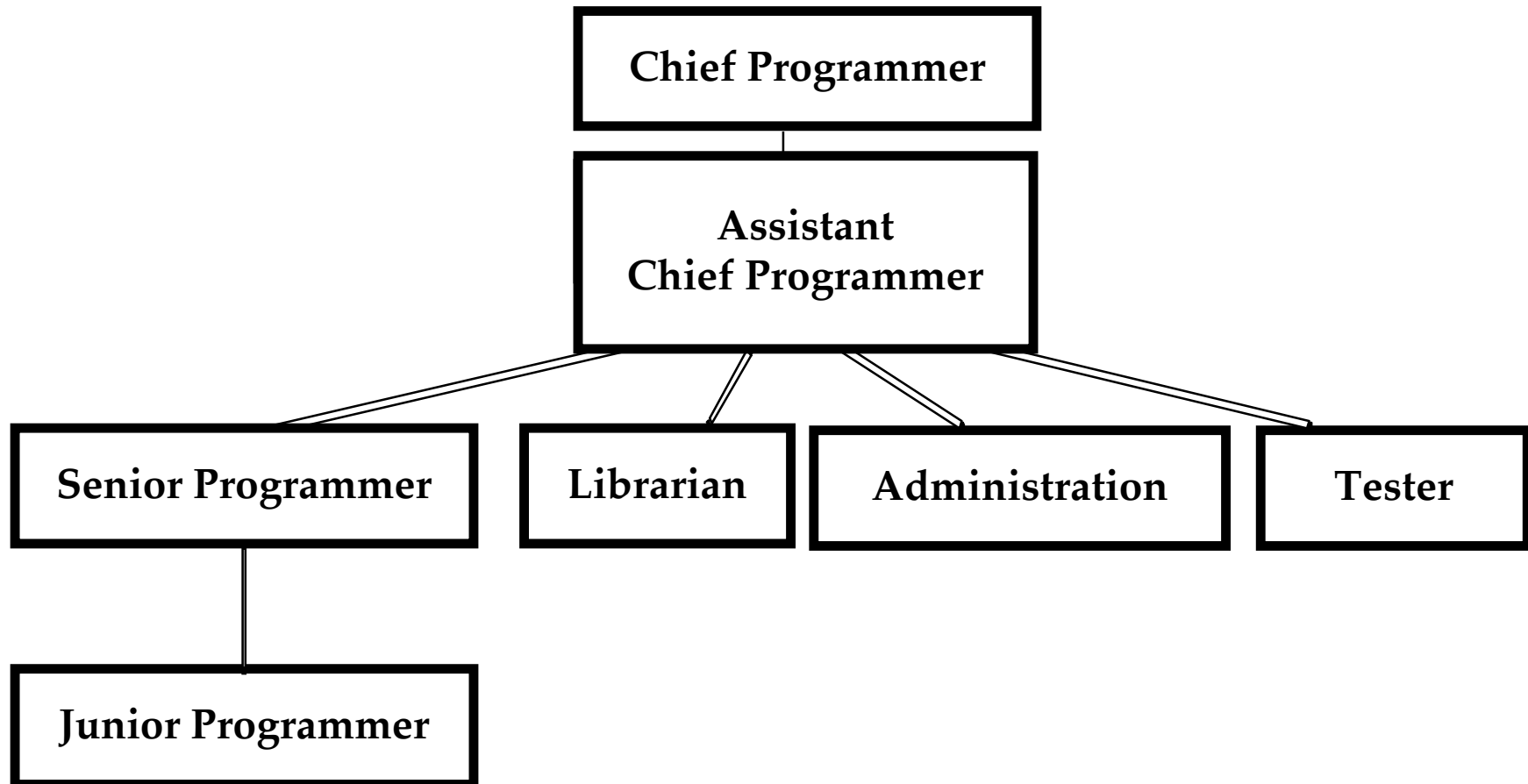


Example of a Functional Organization

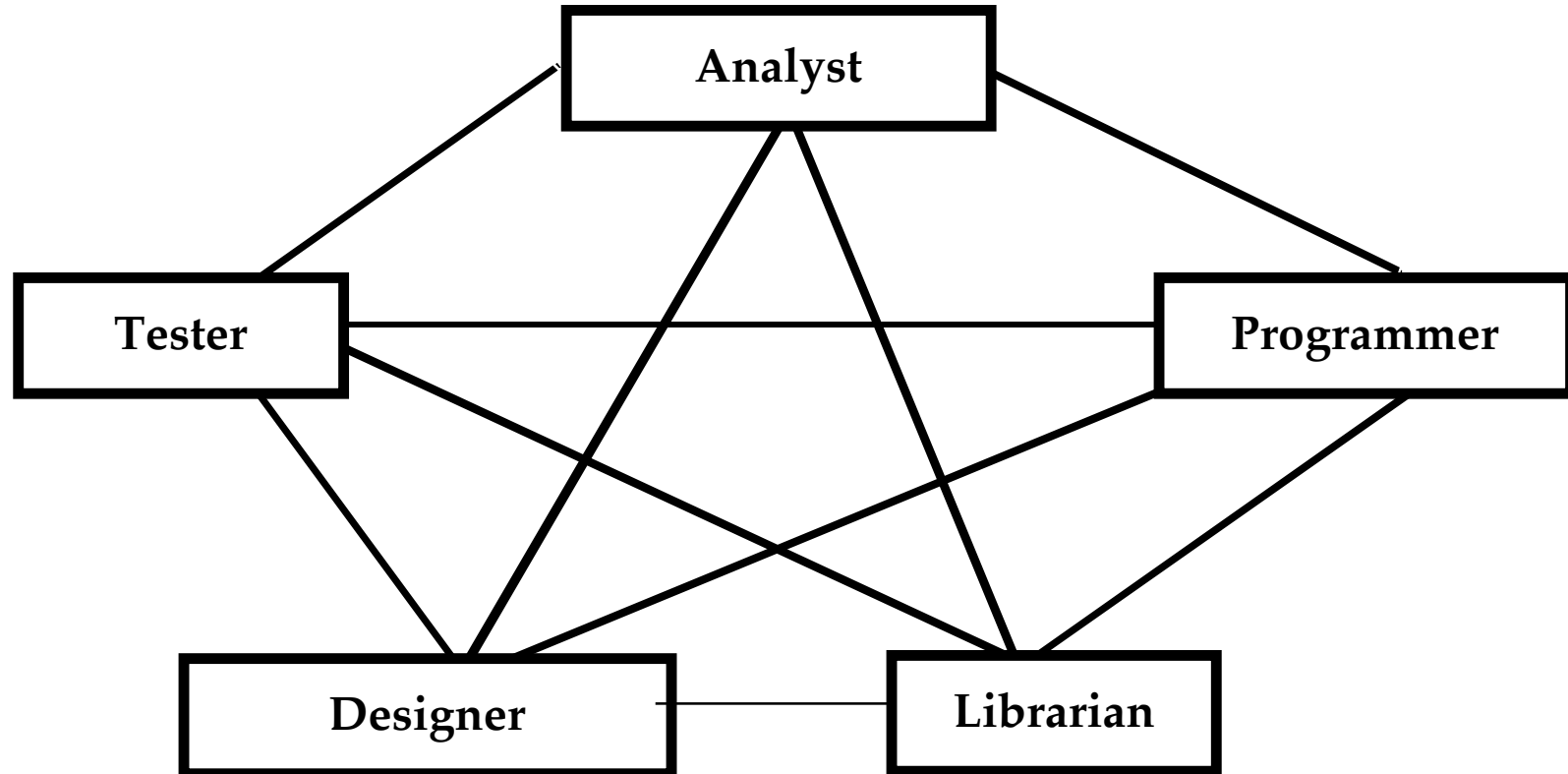


Organization of a „traditional business“

Example of a Hierarchical Organization: Chief Programmer Team [Brooks 1995]



A Nonhierarchical Organization: Egoless Programming [Weinberg 1971]



Responsibilities of the Project Manager

- Determine objectives, schedule and budgets
- Design software project management plan (SPMP)
- Establish focused and motivated teams
- Determine work procedures, reporting systems and communication infrastructure
- Accomplish project objective within time & budget
- Monitor performance against the plan
- Resolve technical and interpersonal conflicts
- Report project activities to upper management
- Keep the client informed and committed
- Contribute to or describe the team members performance approval.

Power Promoter

- Also called executive champion or project champion
- Pushes the change through the existing organizational hierarchy
 - Not necessarily at the top of the organization, but has protection from top level management, otherwise project opponents might be able to prevent the success of the project.
- Tasks:
 - Constantly identify difficulties, resolve issues, and communicate with the project members, especially with the developers.
- Example at project level: Project Leader.
- Example at corporate level: Chief Executive Officer (CEO).

Knowledge Promoter

- Also called the technologist
- Promotes change arising in the application domain or the solution domain. Usually closely associated with the power promoter
- Tasks: Acquire information iteratively, understand the benefits and limitations of new technologies, and argue its adoption with the other developers
- Example at project level: System architect
 - Reports to project manager
 - Has final say over all technical decisions in the system.
- Example at corporate level: Chief Technical Officer (CTO).

Process Promoter

- The process promoter has good knowledge of the communication processes and procedures
- The process promoter is in constant interaction with the power promoter to get consensus on the overall goals
- Tasks: Bridge between the power and knowledge promoter, who often do not speak or understand the same language
- Example at project level: Development lead
 - Responsible for the administrative aspects of a project, including planning, milestones definition, budgeting and communication infrastructure
- Example at corporate level: Chief Information Officer (CIO).



Responsibilities of the Coach

- Listen to gripes from individual team members
- Attend weekly project meetings
- Review weekly team status reports
- Schedule and prepare meetings with project manager
- Insist that project guidelines are followed
- Assign presentations to team members (in-class project meetings, client review, client acceptance test)
- Resolve team member conflicts if they cannot be resolved otherwise

Responsibilities of the Team Leader

- Responsible for intra-team communication (Meeting Management: Primary Facilitator)
 - Run the weekly project meeting
 - Post the agenda before the meeting
 - Define and keep track of action items assigned to team members (who, what, when)
 - Measure progress (Enforce milestones)
 - Deliver work packages for the tasks to the project manager
 - Present team status to project manager
- *Important heuristics: The team leader should be rotated among members of the team.*

Team Leader: Create an Agenda

- Purpose of Meeting
- Desired Outcome
- Information Sharing
- Information Processing
- Meeting Critique

Action Items
(Check Previous Meeting)

Issues
(Check Previous Meeting & BBoards)

New Agenda

Agenda for Database Group

Date: 06/19/96

Location: Primary Facilitator: Bernd Bruegge

Start Time: 12:00 PM Minute Taker:

End Time: 12:00 PM Time Keeper:

Purpose of the Meeting

2. Desired Outcome

3. Information Sharing (15 Minutes)

Include Action Item Text: Yes Update Action Item Text

To exclude Action Items, choose 'No' and press 'Update Action Item Text';
To include Action Items, choose 'Yes' and press 'Update Action Item Text'.
These two fields form a toggle switch for including Action Items in this Agenda.
The 'Update Action Item Text' button can also be used to refresh the Action Items in an Agenda.

4. Information Processing (40 Minutes)

Include Issue Text: Yes Update Issue Text

To exclude Issues, choose 'No' and press 'Update Issue Text';
To include Issues, choose 'Yes' and press 'Update Issue Text'.
These two fields form a toggle switch for including Issues in this Agenda.
The 'Update Issue Text' button can also be used to refresh the set of Issues in an Agenda.

Responsibilities of the API Liaison

- Responsible for inter-team communication
 - API Liaison: Make available public definitions of subsystem developed by the team to the architecture teams (ensure consistency, etc)
 - Coordinate tasks spanning more than one group with other teams
- Responsible for service negotiations.

Responsibilities of the Planner

- Plans the activities of an individual team
- Define project plan for team:
 - Work Breakdown Structure
 - Dependency graph and schedule showing work packages
- Make project plan available to management
- Report team project status to team leader

No explicit planner in many teams
Responsibility usually assumed by team
leaders or project manager.

Responsibilities of the Document Editor

- Collect, proofread and distribute team documentation
- Submit team documentation to documentation team
- Collect agendas
- Take minutes at meetings.

Responsibilities of the Web Master

- Maintain team home page
- Keep track of meeting history
- Keep track of design rationale.

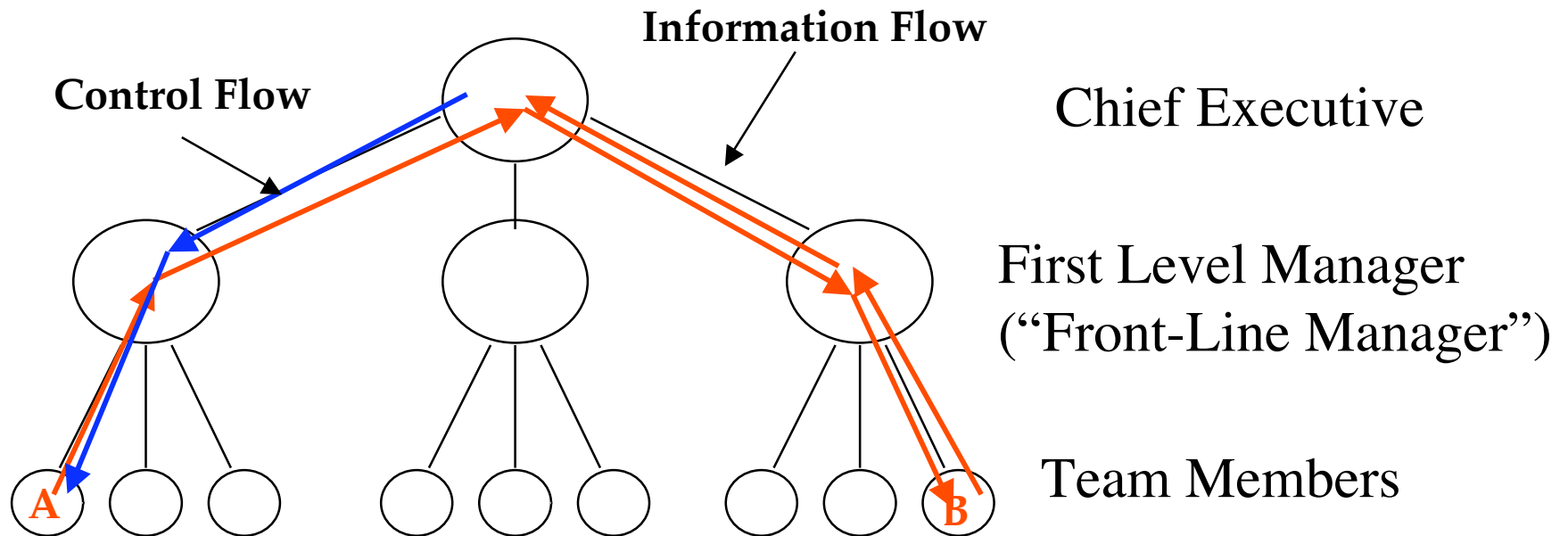
Web Master

- Publish Meeting Information on Team Homepage
 - Should contain agenda, minutes, action items and issues
 - Possibilities:
 - One HTML document per meeting, with anchors (maintained by one role)
 - Separate HTML documents for Agenda, Minutes, etc (maintained by several roles)

Date **Agenda** **Minutes** **Action Items** **Issues**

9/9/96	<u>Agenda</u>	<u>Minutes</u>	<u>Action Items</u>	<u>Issues</u>
9/16/96	<u>Agenda</u>	<u>Minutes</u>	<u>Action Items</u>	<u>Issues</u>

Hierarchical Project Organization

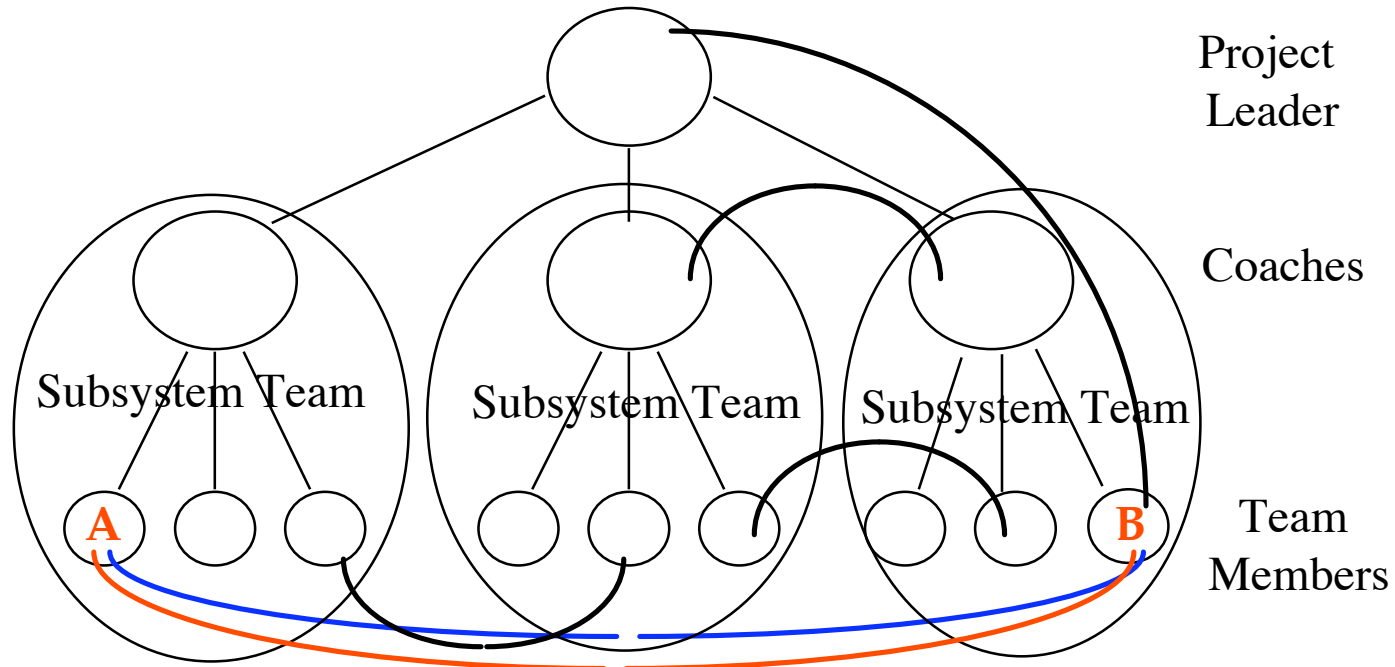


A wants to talk to B: Complicated Information Flow

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

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

Nonhierarchical Project Organization



A wants to talk to B: Communication Flow

B wants to make sure A does a certain change: Decision Flow

Basis of organization:
Nonlinear information flow across dynamically formed units

Heuristics for Project Managers

1. Create team identity

- Clarify team vision and working relationships
- Define team procedures (meeting management, configuration management, system integration strategy)
- Clarify each participant's role
- Make sure the team is functioning

2. Create team member buy-in

- Get commitment to the project goals (hard in matrix organization)
- Get to know other people's style

3. Get support from the environment

- Get a project champion (for example a power promoter)

4. Develop general procedures

- Procedure for conflict resolution
- Procedures for communication between teams and project manager, communication with upper management and for communication with the client.

Micromanagement

- Micromanagement is the excessive involvement of a person in the details of a task assigned to another person
- Micromanagement is inefficient use of the time and energy of all project participants
- It leads to tension and low morale among all project members
- Why do people micro-manage?

Reasons for Micromanagement

- The manager is interested the work and enjoys it
- The manager is a technical expert and feels he/she can do the job best
- The manager may feel the assignment was not explained clearly
- The manager is looking for a way to stay involved with the person or the team
- The manager feels threatened because the managed person has more technical knowledge
- The manager does not have a clear understanding on how to spend project time
- The manager wants to stay up-to-date in case somebody else asks about the work.

Overcoming Micro Management

- Don't be defensive when the manager asks questions
 - Doing so make it appear as if you are hiding something and the manager will worry even more
- Thank the micromanager for the interest and time
 - Complaining about micromanagement will cause the micromanager to do it even more
- Offer to explain to the micromanager how you will approach your tasks
- Work with the micromanager to develop a scheme for sharing progress and accomplishments.

Other Project Lists

- **Stakeholder list**
 - Identifies people and groups who support or are affected by your project
 - This list does not include people outside of the organization or those who are merely interested in the project
- **Distribution Lists**
 - Identifies people who receive copies of written project communication.
 - The presence of people on distribution lists does not ensure that they actually support the project (Often out of date)
- **Team member lists**
 - People whose work is directed by the project manager.

Linear Responsibility Chart

- A **linear responsibility chart** is a matrix that depicts the role that each project participant will play in different activities identified in the work breakdown structure.
- Rows: Project activities
- Columns: Roles/Project participants
- Entries: Type of responsibility
 - *P (Primary responsibility)*: You have committed to ensure that the desired result is achieved
 - *S (Secondary responsibility)*: You have committed to some portion of the result
 - *A (Approval)*: You are not doing the work, but you will approve what has been done
 - *R (Review)*: You will review and comment on the work product of an activity
 - *O (Output)*: You will receive the workproduct of an activity
 - *I (Input)*: You will provide input for a task or activity

Example of a Responsibility Chart

	Project Manager	Team Leader	Team Member A	Team Member B
Develop SPMP	P			
Run weekly meeting		A	P	S
Write SDD	P	S	S	S
	<p><i>Legend:</i> P = Primary responsibility S = Secondary responsibility) A = Approval</p>			

Another Example of a Responsibility Chart

	Project Manager	Team Leader	Team Member A	Team Member B
Develop SPMP	A		P	S

- The Project Manager has delegated the SPMP to Team Member A
- The delegation bypasses the team leader
Is that a problem?
- Team Member B helps by writing a section.