Managing people

- Managing people working as individuals and in groups

Objectives

- To emphasize the importance of people in the software engineering process.
- To describe fundamental cognitive factors which managers should be aware of.
- To discuss group working and group organization.
- To explain why the working environment for software engineers is important.

Topics covered

- Cognitive fundamentals
- Management implications
- Project staffing
- Group working
- Working environments

People in the process

- People are an organization’s most important assets.
- The tasks of a manager are essentially people oriented. Unless there is some understanding of people, management will be unsuccessful.
- Software engineering is primarily a cognitive activity.

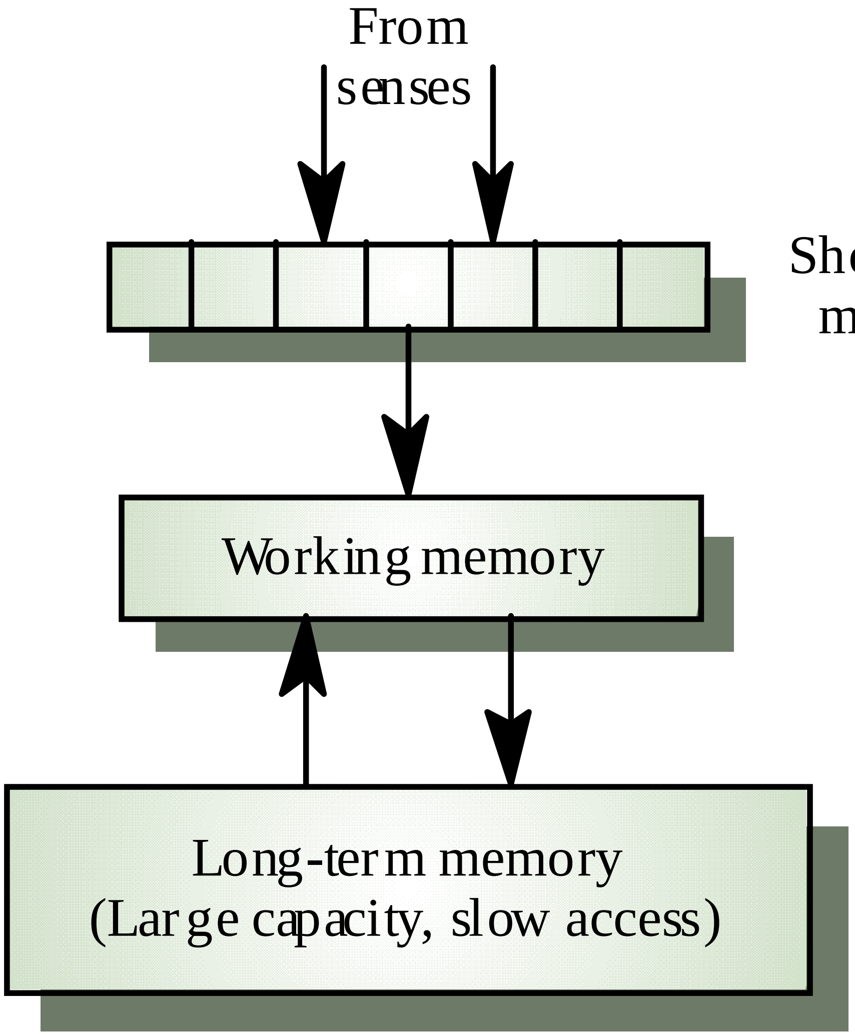
Management activities

- Problem solving (using available people)
- Motivating (people who work on a project)
- Planning (what people are going to do)
- Estimating (how fast people will work)
- Controlling (people’s activities)
- Organizing (the way in which people work)

Cognitive fundamentals

- People don’t all think the same way but everyone is subject to some basic constraints on their thinking due to:
  - Memory organization
  - Knowledge representation
  - Motivation influences
- If we understand these constraints, we can understand how they affect people participating in the software process.

Memory organization



Short-term memory

- Fast access, limited capacity
- 5-7 locations
- Holds “chunks” of information where the size of a chunk may vary depending on its familiarity.
- Fast decay time

Working memory

- Larger capacity, longer access time.
- Memory area used to integrate information from short-term memory and long-term memory.
- Relatively fast decay time.

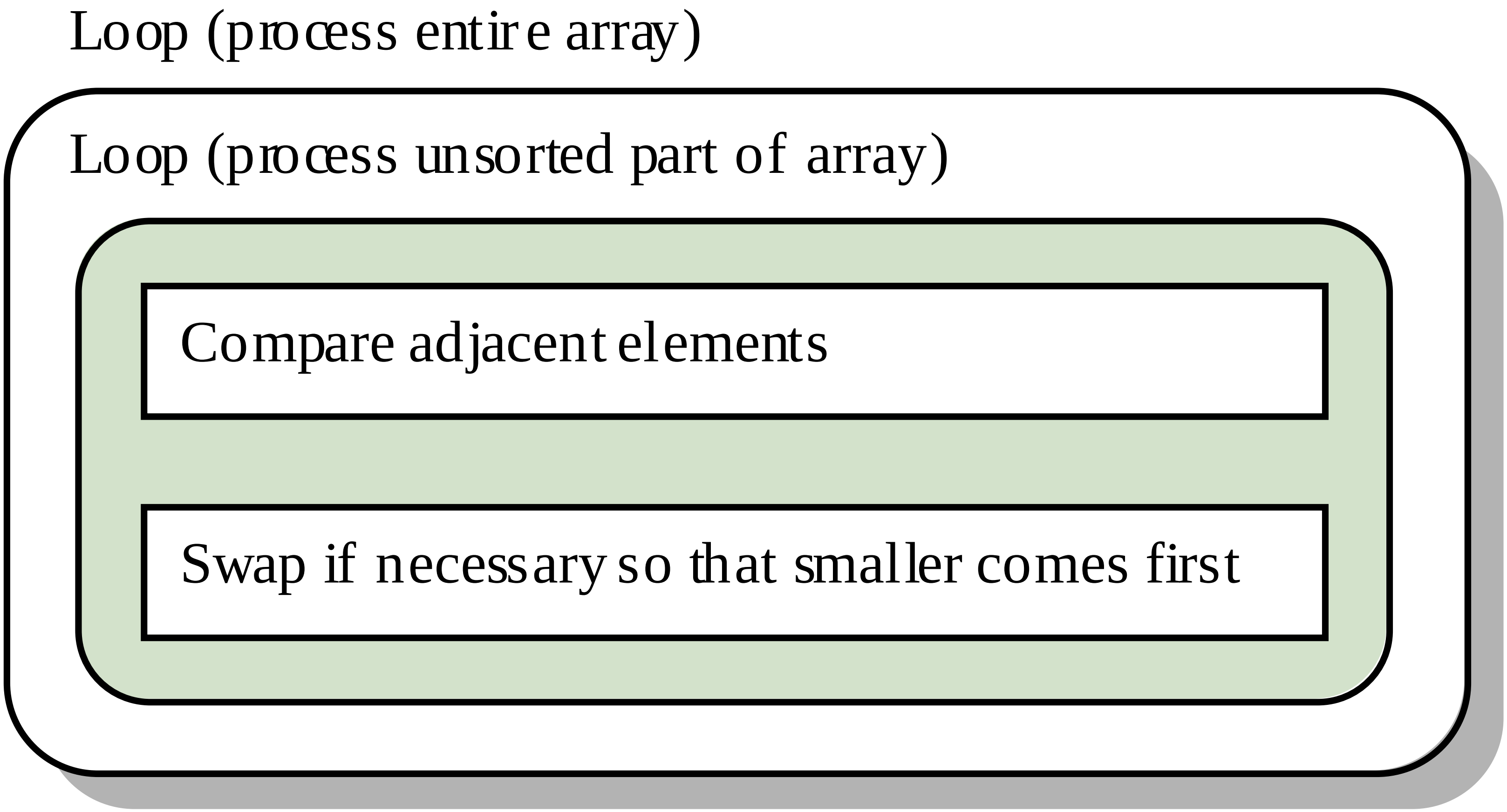
Long-term memory

- Slow access, very large capacity.
- Unreliable retrieval mechanism.
- Slow but finite decay time - information needs reinforcing.
- Relatively high threshold - work has to be done to get information into long-term memory.

Information transfer

- Problem solving usually requires transfer between short-term memory and working memory.
- Information may be lost or corrupted during this transfer.
- Information processing occurs in the transfer from short-term to long-term memory.

Cognitive chunking



Loop (process entire array)

Loop (process unsorted part of array)

Compare adjacent elements

Swap if necessary so that smaller comes first

Knowledge modelling

- **Semantic knowledge:** knowledge of concepts such as the operation of assignment, concept of parameter passing etc.
- **Syntactic knowledge:** knowledge of details of a representation *e.g.*, a Java while loop.

Knowledge acquisition

- Semantic knowledge is acquired through experience and active learning - the “ah” factor
- Syntactic knowledge is acquired by memorization.
- New syntactic knowledge can interfere with existing syntactic knowledge.
  - *E.g.*, problems in mixing up syntax of different programming languages

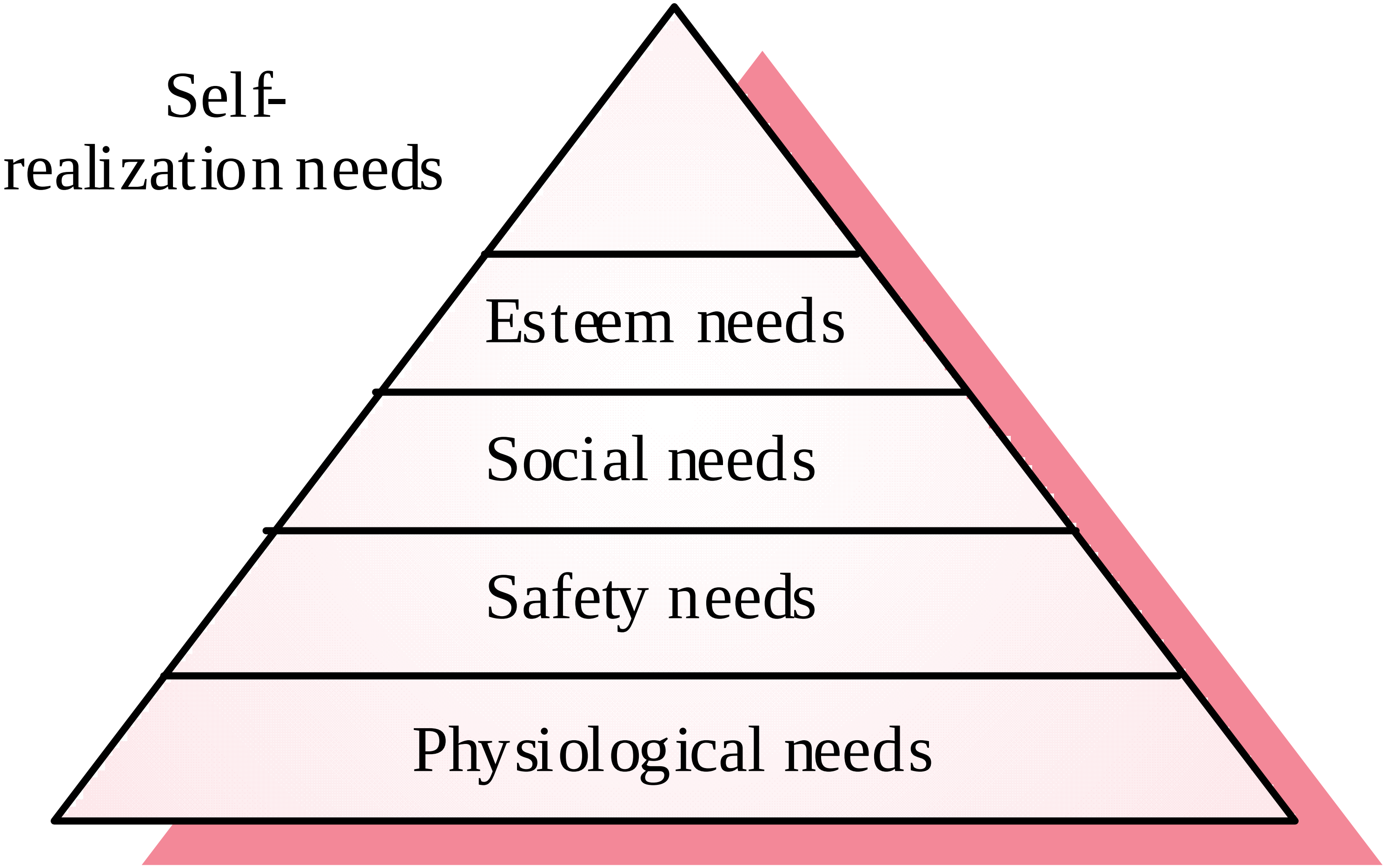
Semantic knowledge

- *E.g.*, **Computing concepts:**
  - notion of a writable store
  - iteration
  - concept of an object
  - ...

Motivation

- An important role of a manager is to motivate the people working on a project.
- Motivation is a complex issue but it appears that their are different types of motivation based on:
  - Basic needs (*e.g.*, food, sleep)
  - Personal needs (*e.g.*, respect, self-esteem)
  - Social needs (*e.g.*, to be accepted as part of a group)

Human needs hierarchy



Personality types

- *Needs hierarchy* is an over-simplification.
- Must also take into account different personality types:
  - Task-oriented
  - Self-oriented
  - Interaction-oriented

Personality types

- Task-oriented.
  - The motivation for doing the work is the work itself.
- Self-oriented.
  - The work is a means to an end which is the achievement of individual goals - *e.g.*, to get rich, to play tennis, to travel etc.
- Interaction-oriented
  - The principal motivation is the presence and actions of co-workers. People go to work because they like to go to work.

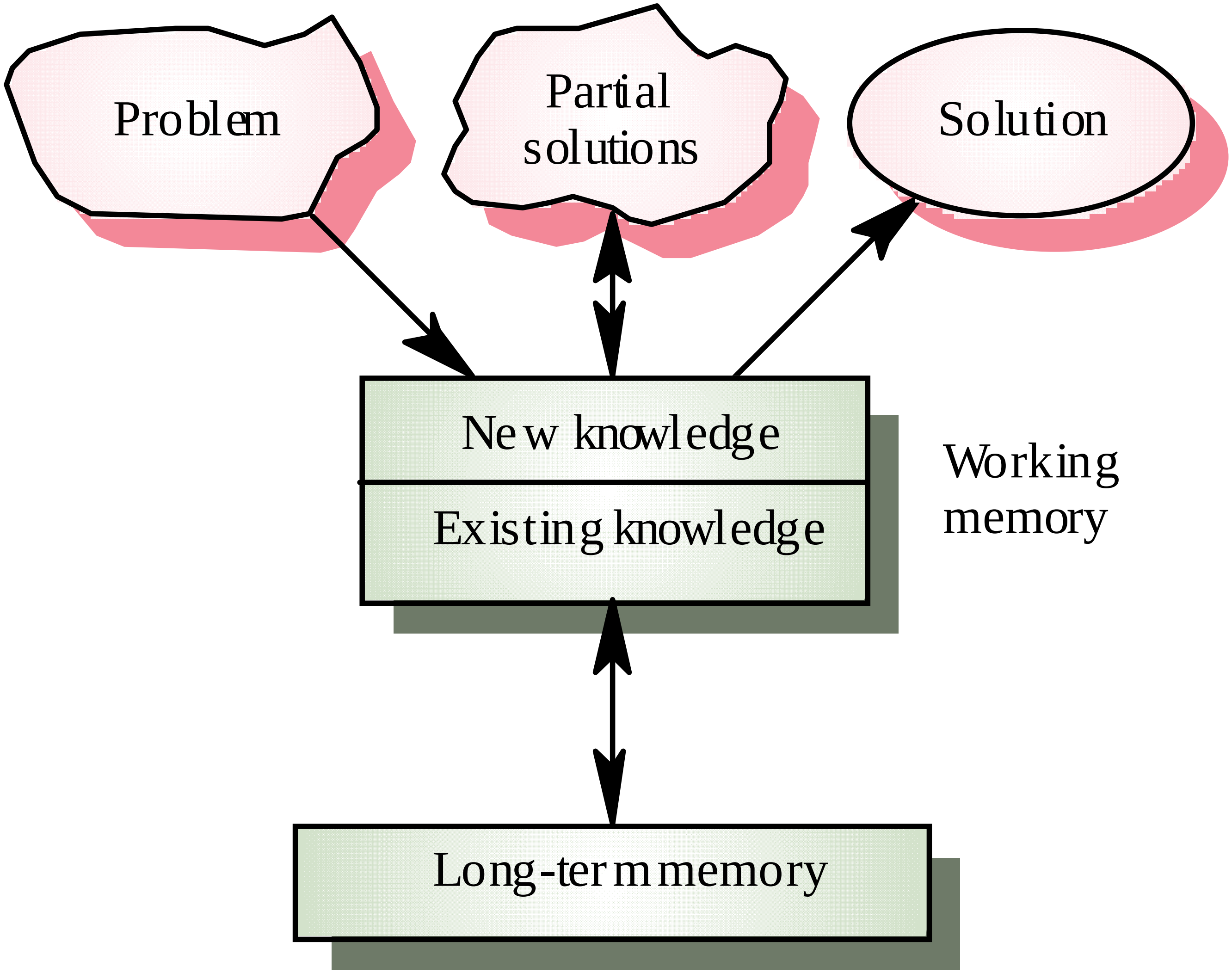
Motivation balance

- Individual motivations are made up of elements of each class.
- Balance can change depending on personal circumstances and external events.

Problem solving

- **Structured programming:** Limited control structures place a lesser load on short-term and working memory.
- **Software development ability:** is the ability to integrate new knowledge with existing computer and task knowledge and hence derive creative problem solutions.
- Thus, problem solving is language independent.

Problem solving



Training

- Language concepts should match semantic structures formed - more errors are likely to be made by low-level language programmers.
- Easy for programmers to learn a new language of the same type.
- May be more difficult for experienced programmers to learn a new language of a different type than for novices.

Motivation

- Motivations depend on satisfying needs.
- It can be assumed that physiological and safety needs are satisfied.
- Social, esteem and self-realization needs are most significant from a managerial viewpoint.

Need satisfaction

- Social
  - Provide communal facilities
  - Allow informal communications
- Esteem
  - Recognition of achievements
  - Appropriate rewards
- Self-realization
  - Training - people want to learn more
  - Responsibility

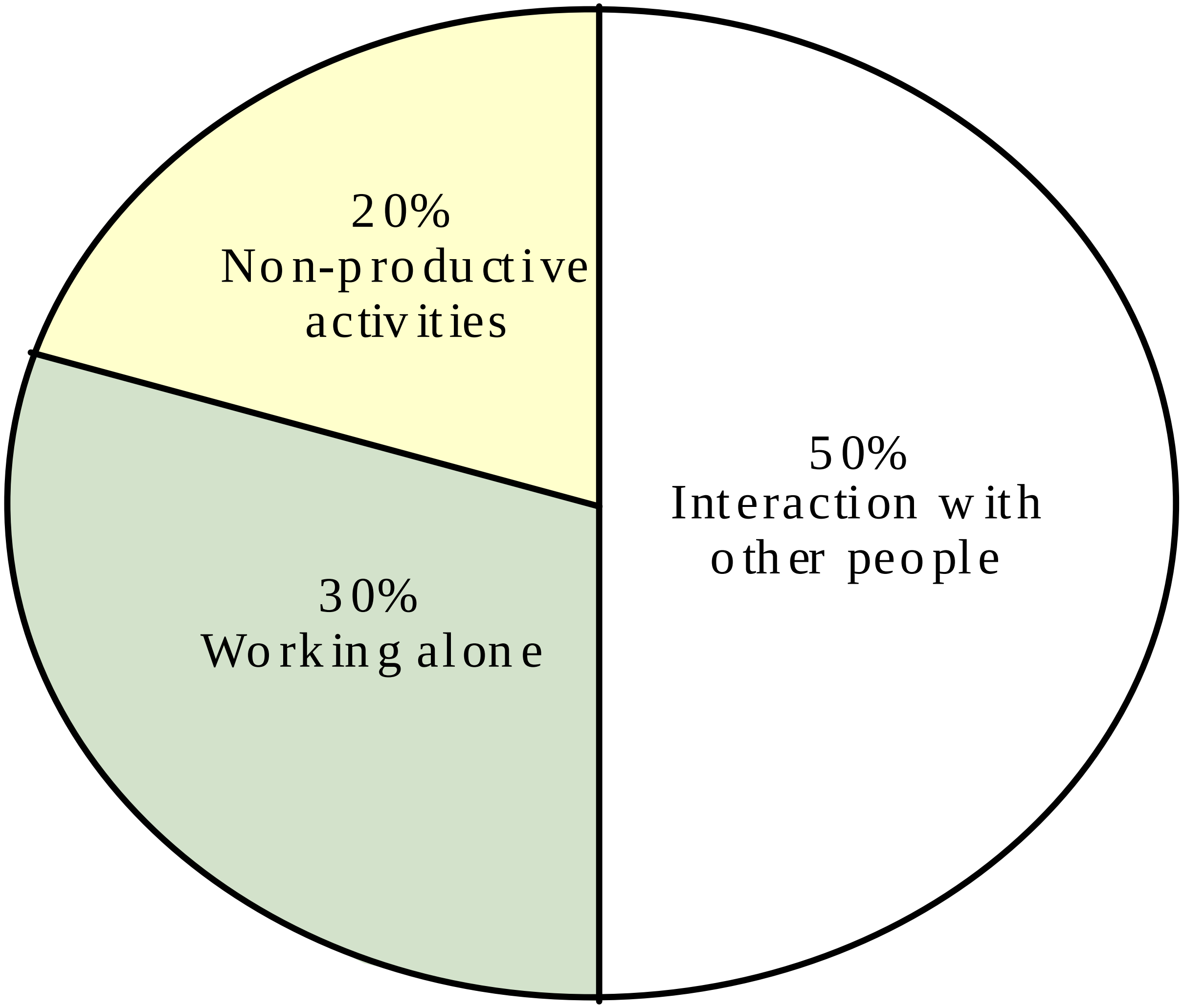
Project staffing

- Choosing people to work on a project is a major managerial responsibility.
- Appointment decisions are usually based on:
  - information provided by the candidate (their CV)
  - information gained at an interview
  - recommendations from other people who know the candidate
- Some companies use aptitude tests
  - There is no agreement on whether or not these tests are actually useful

Group working

- Software engineering is a group activity
  - The development schedule for most non-trivial software projects is such that they cannot be completed by one person working alone.
  - Group interaction is a key determinant of group performance.
- Flexibility in group composition is limited
  - Managers must do the best they can with available people

Time distribution



Group composition

- Group composed of members who share the same motivation can be problematic:
  - Task-oriented - everyone wants to do their own thing.
  - Self-oriented - everyone wants to be the boss.
  - Interaction-oriented - too much chatting, not enough work.
- An effective group has a balance of all types.
- Can be difficult to achieve because most engineers are task-oriented.

Group leadership

- Leadership depends on respect not title.
- There should be both a technical and an administrative leader.
- Democratic leadership is more effective that autocratic leadership.
- A career path based on technical competence should be supported.

Group cohesiveness

- In a cohesive group, members consider the group to be more important than any individual in it.
- Advantages of a cohesive group are:
  - Group quality standards can be developed.
  - Team members learn from each other and get to know each other’s work.
  - Egoless programming where members strive to improve each other’s programs can be practised.

Group loyalties

- Group members tend to be loyal to cohesive groups.
- “Groupthink” is preservation of group irrespective of technical or organizational considerations.
- Management should act positively to avoid groupthink by forcing external involvement with each group.

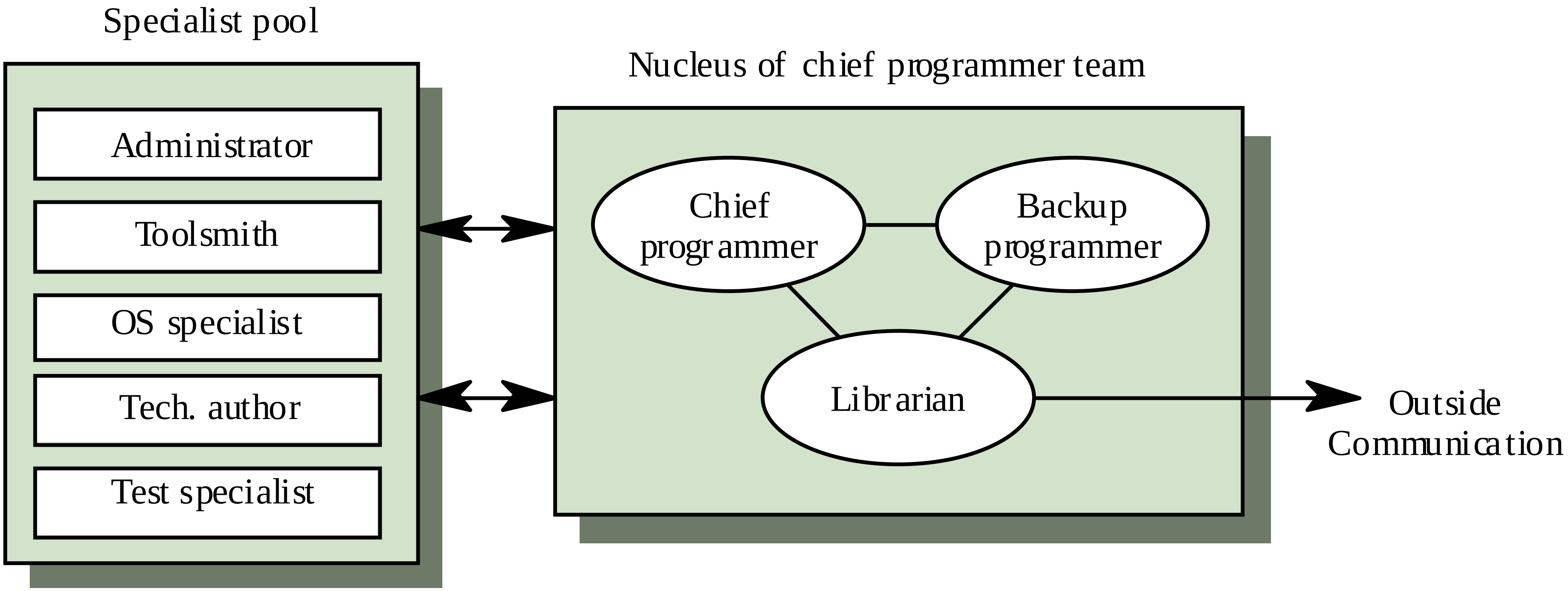
Group communications

- Status of group members
  - Higher status members tend to dominate conversations.
- Personalities in groups
  - Too many people of the same personality type can be a problem.
- Communication channels
  - Communication channelled though a central coordinator tend to be ineffective.

Team organization

- Software engineering group sizes should be relatively small (< 8 members).
- Break big projects down into multiple smaller projects.
- Small teams may be organized in an informal, democratic way.
- Chief programmer teams try to make the most effective use of skills and experience.

Chief programmer teams



Chief programmer teams

- Consist of a kernel of specialists helped by others added to the project as required.
- Reportedly successful but problems with this approach are:
  - Finding talented chief programmers.
  - Disruption to normal organizational structures.
  - Demotivating effect on those who are not chief programmers.

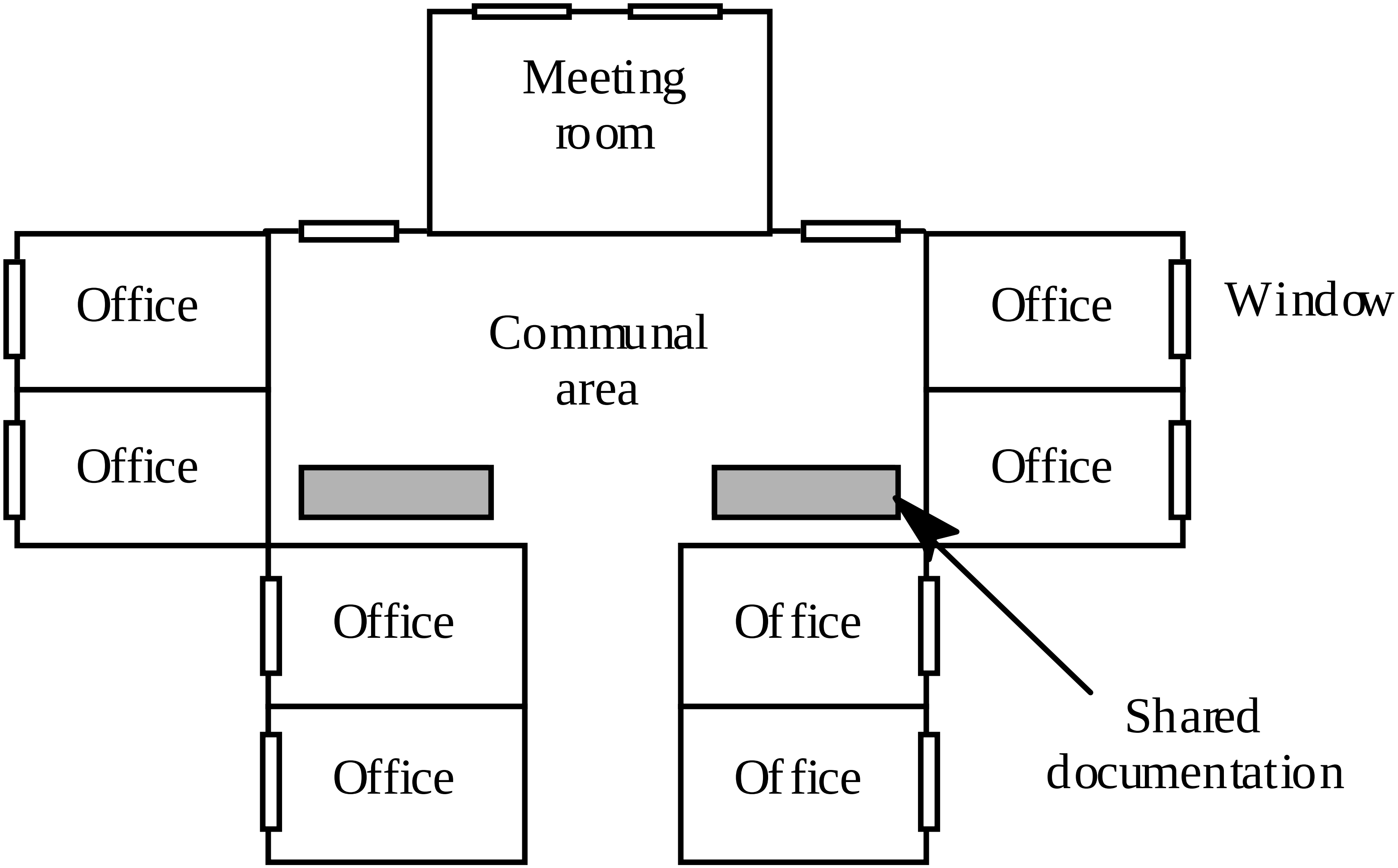
Working environments

- Physical workplace provision has an important effect on individual productivity and satisfaction:
  - Comfort
  - Privacy
  - Facilities
- Health and safety considerations must be taken into account:
  - Lighting
  - Heating
  - Furniture

Environmental factors

- Privacy - each engineer requires an area for uninterrupted work.
- Outside awareness - people prefer to work in natural light.
- Personalization - individuals adopt different working practices and like to organize their environment in different ways.

Office grouping



Equipment provision

- Computers
  - Each engineer should have a personal workstation with access to software which is needed for his/her job.
- Telecommunications
  - telephone, fax, e-mail, networking facilities.
  - groupware - systems such as Lotus Notes facilitate information sharing.
  - Good telecommunications facilities can reduce travel costs drastically.

Key points

- Managers must have some understanding of people if they are to be effective.
- Human cognitive structures place fundamental limits on the ways in which people work.
- The success of techniques such as structured programming and OO development is partially a consequence of cognitive structures.
- Staff selection factors include education, domain experience, adaptability and personality.

Key points

- Software development groups should be small and cohesive.
- Group communications are affected by status, size, and organization of the group.
- The working environment has a significant effect on productivity.