



# IT3205: Fundamentals of Software Engineering (Compulsory)

**BIT – 2<sup>nd</sup> Year**  
**Semester 3**



# IT3205: Fundamentals of Software Engineering

## Software Project Management

Duration: 3 hours



## Learning Objectives

- State the requirement of managerial control of the development process.
- Describe the main phases of software project management.
- Describe project planning and project scheduling activities in detail.



# Detailed Syllabus

- 8.1 Need for the proper management of software projects
- 8.2 Management activities
  - 8.2.1 Project planning
  - 8.2.2 Estimating costs
  - 8.2.3 Project scheduling
  - 8.2.4 Risk management
  - 8.2.5 Managing people



# 8.1 NEED FOR THE PROPER MANAGEMENT OF SOFTWARE PROJECTS





# Software Project Management

- Concerned with the activities involved in ensuring that software is delivered on time and on schedule and in accordance with the requirements of the organizations developing and procuring the software.
- Project management is needed because software development is always subject to budget and schedule constraints that are set by the organization developing the software.



## Success Criteria

- Deliver the software to the customer at the agreed time
- Keep overall costs within budget
- Deliver software that meets the customer's expectations
- Maintain a happy and well-functioning development team



# Software Management Distinctions

- The product is intangible
  - Software cannot be seen or touched. Software project managers cannot see progress by simply looking at the artefact that is being constructed
- Many software projects are 'one-off' projects
  - Large software projects are usually different in some ways from previous projects. Even managers who have lots of previous experience may find it difficult to anticipate problems
- Software processes are variable and organization specific
  - We still cannot reliably predict when a particular software process is likely to lead to development problems

## 8.2 MANAGEMENT ACTIVITIES





# SPM Activities

1. Project Planning
2. Cost Estimation
3. Project Scheduling
4. Risk Management
5. Team Management



# 1. Project Planning

- Probably the most time consuming project management activity
- This is a continuous activity from initial concept through to system delivery
- Plans must be regularly revised as new information becomes available
- Various different types of plans may be developed to support the main software project plan that is concerned with schedule and budget



# Contents of a Project Plan

- **Introduction**

- This briefly describes the objectives of the project and sets out the constraints (i.e. budget, schedule, ... etc.) which affects the project management

- **Project organization**

- This describes the way in which the development team is organized, the people involved and their roles in the team

- **Risk analysis**

- This describes possible project risks, the likelihood of these risks arising and the risk reduction strategies which are proposed



# Contents of a Project Plan

- **Hardware & software requirements**
  - Describes the hardware and the support software required to carry out the development
- **Work breakdown**
  - Describes the breakdown of the project into activities and identifies the milestones and deliverables associated with each activity
- **Project schedule**
  - This describes the dependencies between activities, the estimated time required to reach each milestone and the allocation of people to activities



# Contents of a Project Plan

- **Monitoring and reporting mechanisms**
  - This describes the management reports which should be produced, when these should be produced and the project monitoring mechanisms used
- **Reference to quality plan**
  - Quality procedures for the project must be specified in a quality plan for the project



## 2. Software Cost Estimation

- Cost estimation is concerned with predicting the resources required for a software development process
- Fundamental estimation questions
  1. How much effort is required to complete an activity?
  2. How much calendar time is needed to complete an activity?



# Software Cost Components

- Hardware and software costs
- Travel and training costs
- Effort costs
  - Salaries of engineers
  - Social and insurance costs
- Other overheads
  - Costs of logging, heating and lighting
  - Costs of networking and communications
  - Costs of other facilities



## Costing & Pricing

- Estimates are made to discover the cost of the developer to produce software
- There is not a simple relationship between the development cost and the price charged to the customer
- Broader organizational, economic, political and business considerations influence the price charged



# Productivity Measures

- **Size related measures**
  - based on some output from the software process
    - For example lines of delivered source code
- **Function related measures**
  - based on an estimate of the functionality of the delivered software
  - Function Points (FP) are the best known of this type of measure



# Cost Estimation Techniques

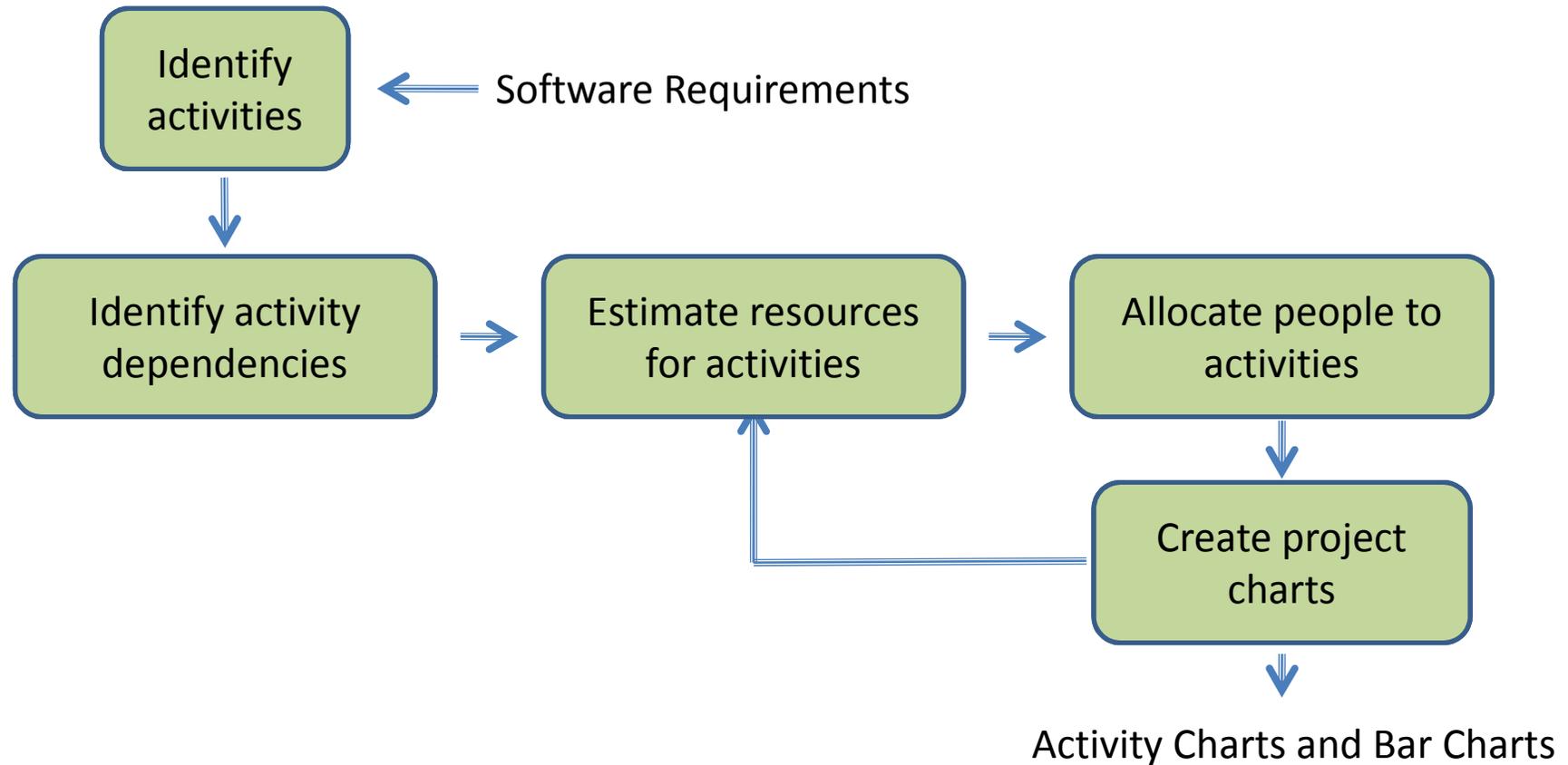
- Expert judgment
  - Use expert knowledge
- Estimation by analogy
  - Use past similar projects to do the estimation
- Function point analysis
  - Uses the requirements specification to assess inputs, outputs, file accesses, user interactions and interfaces and calculates the size based on these
- Algorithmic cost modeling
  - Cost is estimated as a mathematical function of product and process attributes



## 3. Project Scheduling

- Split project into tasks and estimate time and resources required to complete each task
- Organize tasks concurrently to make optimal use of workforce
- Minimize task dependencies to avoid delays caused by one task waiting for another to complete

# The Project Scheduling Process





## Problems in Project Scheduling

- Estimating the difficulty of problems and hence the cost of developing a solution.
- Productivity is not proportional to the number of people working on a task.
- Adding people to a late project makes it later because of communication overheads.
- The unexpected always happens therefore allow contingency in planning.



## Bar Charts & Activity Networks

- These are graphical notations used to illustrate the project schedule.
- They show project breakdown into tasks.
- Activity networks show task dependencies and the critical path.
- Bar charts show schedule against calendar time.



## 4. Risk Management

- Risk management is concerned with identifying risks and drawing up plans to minimize their effect on a project.
- A risk is a probability that some adverse circumstance will occur.
- The results of the risk analysis should be documented in the project plan along with the analysis of the consequences of risks occurring.



# Risk Categories

- **Project risks**
  - affect the project schedule or resources
- **Product risks**
  - affect the quality or the performance of the software being developed
- **Business risks**
  - affect the organization developing or procuring the software



# Examples

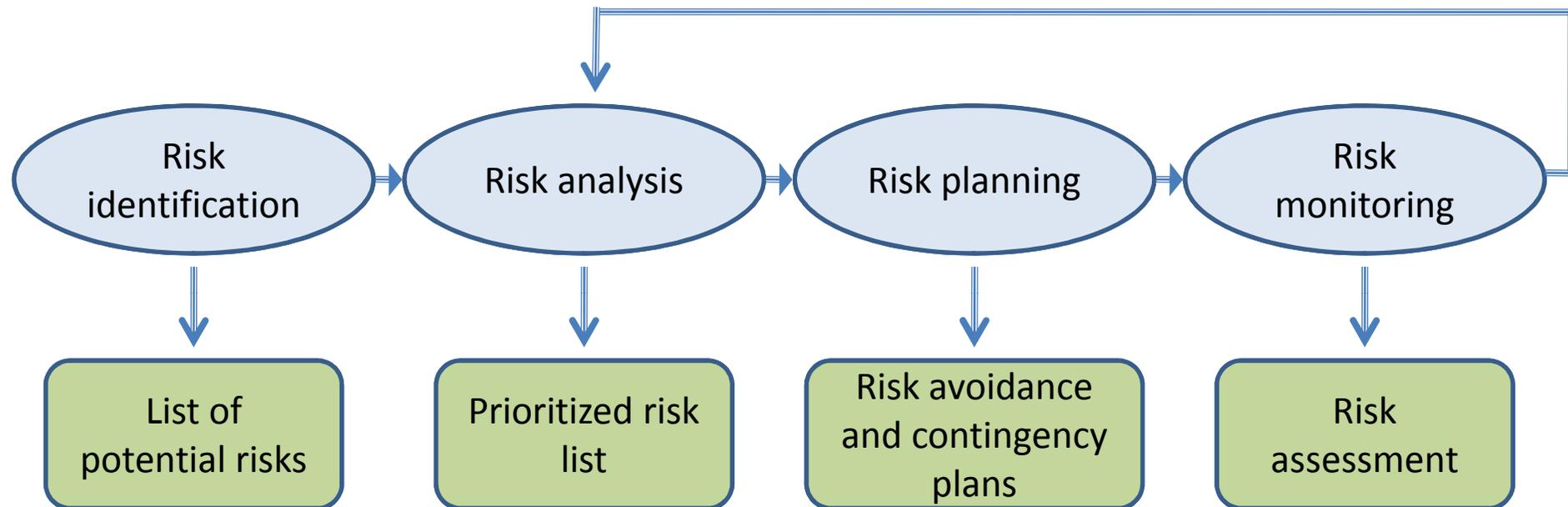
Risk	Affects	Description
Staff turnover	Project	Experienced staff will leave the project before it is finished.
Management change	Project	There will be a change of organizational management with different priorities.
Hardware unavailability	Project	Hardware that is essential for the project will not be delivered on schedule.
Requirements change	Project and product	There will be a larger number of changes to the requirements than anticipated.
Specification delays	Project and product	Specifications of essential interfaces are not available on schedule.
Size underestimate	Project and product	The size of the system has been underestimated.
CASE tool underperformance	Product	CASE tools, which support the project, do not perform as anticipated.
Technology change	Business	The underlying technology on which the system is built is superseded by new technology.
Product competition	Business	A competitive product is marketed before the system is completed.



# Risk Management Process

- Risk identification
  - Identify project, product and business risks;
- Risk analysis
  - Assess the likelihood and consequences of these risks;
- Risk planning
  - Draw up plans to avoid or minimise the effects of the risk;
- Risk monitoring
  - Monitor the risks throughout the project;

# Risk Management Process





# Risk Identification

- May be a team activities or based on the individual project manager's experience.
- A checklist of common risks may be used to identify risks in a project.
- Types of such risks are;
  - Technology risks
  - People risks
  - Organizational risks
  - Tool risks
  - Requirements risks
  - Estimation risks



# Risk Types

Risk type	Possible risks
Technology	The database used in the system cannot process as many transactions per second as expected. (1) Reusable software components contain defects that mean they cannot be reused as planned. (2)
People	It is impossible to recruit staff with the skills required. (3) Key staff are ill and unavailable at critical times. (4) Required training for staff is not available. (5)
Organizational	The organization is restructured so that different management are responsible for the project. (6) Organizational financial problems force reductions in the project budget. (7)
Tools	The code generated by software code generation tools is inefficient. (8) Software tools cannot work together in an integrated way. (9)
Requirements	Changes to requirements that require major design rework are proposed. (10) Customers fail to understand the impact of requirements changes. (11)
Estimation	The time required to develop the software is underestimated. (12) The rate of defect repair is underestimated. (13) The size of the software is underestimated. (14)



# Risk Indicators

Risk type	Potential indicators
Technology	Late delivery of hardware or support software; many reported technology problems.
People	Poor staff morale; poor relationships amongst team members; high staff turnover.
Organizational	Organizational gossip; lack of action by senior management.
Tools	Reluctance by team members to use tools; complaints about CASE tools; demands for higher-powered workstations.
Requirements	Many requirements change requests; customer complaints.
Estimation	Failure to meet agreed schedule; failure to clear reported defects.



# Risk Analysis

- Assess probability and seriousness of each risk
- Probability may be very low, low, moderate, high or very high
- Risk consequences might be catastrophic, serious, tolerable or insignificant



# Risk Analysis - Examples

Risk	Probability	Effects
Organizational financial problems force reductions in the project budget (7).	Low	Catastrophic
It is impossible to recruit staff with the skills required for the project (3).	High	Catastrophic
Key staff are ill at critical times in the project (4).	Moderate	Serious
Faults in reusable software components have to be repaired before these components are reused. (2).	Moderate	Serious
Changes to requirements that require major design rework are proposed (10).	Moderate	Serious
The organization is restructured so that different management are responsible for the project (6).	High	Serious
The database used in the system cannot process as many transactions per second as expected (1).	Moderate	Serious



# Risk Analysis - Examples

Risk	Probability	Effects
The time required to develop the software is underestimated (12).	High	Serious
Software tools cannot be integrated (9).	High	Tolerable
Customers fail to understand the impact of requirements changes (11).	Moderate	Tolerable
Required training for staff is not available (5).	Moderate	Tolerable
The rate of defect repair is underestimated (13).	Moderate	Tolerable
The size of the software is underestimated (14).	High	Tolerable
Code generated by code generation tools is inefficient (8).	Moderate	Insignificant



## Risk Planning

- Consider each risk and develop a strategy to manage that risk
- Avoidance strategies
  - The probability that the risk will arise is reduced
- Minimisation strategies
  - The impact of the risk on the project or product will be reduced
- Contingency plans
  - If the risk arises, contingency plans are plans to deal with that risk



# Risk Monitoring

- Assess each identified risks regularly to decide whether or not it is becoming less or more probable.
- Also assess whether the effects of the risk have changed.
- Each key risk should be discussed at management progress meetings.



# Risk Management Strategies

Risk	Strategy
Organizational financial problems	Prepare a briefing document for senior management showing how the project is making a very important contribution to the goals of the business and presenting reasons why cuts to the project budget would not be cost-effective.
Recruitment problems	Alert customer to potential difficulties and the possibility of delays; investigate buying-in components.
Staff illness	Reorganize team so that there is more overlap of work and people therefore understand each other's jobs.
Defective components	Replace potentially defective components with bought-in components of known reliability.
Requirements changes	Derive traceability information to assess requirements change impact; maximize information hiding in the design.



# Risk Management Strategies

Risk	Strategy
Organizational restructuring	Prepare a briefing document for senior management showing how the project is making a very important contribution to the goals of the business.
Database performance	Investigate the possibility of buying a higher-performance database.
Underestimated development time	Investigate buying-in components; investigate use of a program generator.



## 5. Team Management

- People are an organisation's most important assets.
- The tasks of a manager are essentially people-oriented.
- Unless there is some understanding of people, management will be unsuccessful.
- Poor people management is an important contributor to project failure.



# Team Management

- Most professional software is developed by project teams ranging in size from two to several hundred people.
- Each group is responsible for a sub project.
- As a general rule, software engineering project groups should normally have no more than eight members.



# People Management Factors

- **Consistency**
  - Team members should all be treated in a comparable way without favourites or discrimination.
- **Respect**
  - Different team members have different skills and these differences should be respected.
- **Inclusion**
  - Involve all team members and make sure that people's views are considered.
- **Honesty**
  - You should always be honest about what is going well and what is going badly in a project.



## Selecting Staff

- An important project management task is team selection.
- Information on selection comes from:
  - Information provided by the candidates.
  - Information gained by interviewing and talking with candidates.
  - Recommendations and comments from other people who know or who have worked with the candidates.



# Motivating People

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



## Personality Types

- The needs hierarchy is almost certainly an oversimplification of motivation in practice.
- Motivation should 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.



# Teamwork

- Most 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
- A good group is cohesive and has a team spirit. The people involved are motivated by the success of the group as well as by their own personal goals.
- 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



# Factors that influence group working

- **Group composition**
  - Is the group composed of the right balance of skills, experience and personalities?
- **Group cohesiveness**
  - Does the group think of itself as a team rather than as a collection of individuals who are working together?
- **Group communications**
  - Do group members communicate with each other effectively?
- **Group organization**
  - Is the team organized in such a way that everyone feels valued and is satisfied with their role in the group?



# 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
- This can be difficult to achieve as software engineers are often task-oriented
- Interaction-oriented people are very important as they can detect and defuse tensions that arise



# Group Cohesiveness

- In a cohesive group, members consider the group to be more important than any individual in it.
- The advantages of a cohesive group are:
  - Group quality standards can be developed by the group members.
  - Team members learn from each other and get to know each other's work; Inhibitions caused by ignorance are reduced.
  - Knowledge is shared. Continuity can be maintained if a group member leaves.
  - Refactoring and continual improvement is encouraged. Group members work collectively to deliver high quality results and fix problems, irrespective of the individuals who originally created the design or program.



# Group Communications

- Good communications are essential for effective group working.
- Information must be exchanged on the status of work, design decisions and changes to previous decisions.
- Good communications also strengthens group cohesion as it promotes understanding.



# Group Communications

- Factors that affect to the group communications are;
  - Group size
    - The larger the group, the harder it is for people to communicate with other group members
  - Group structure
    - Communication is better in informally structured groups than in hierarchically structured groups
  - Group composition
    - Communication is better when there are different personality types in a group and when groups are mixed rather than single sex.
  - The physical work environment
    - Good workplace organisation can help encourage communications



# Group Organization

- The way that a group is organized affects the decisions that are made by that group, the ways that information is exchanged and the interactions between the development group and external project stakeholders
  - Key questions include:
    - Should the project manager be the technical leader of the group?
    - Who will be involved in making critical technical decisions, and how will these be made?
    - How will interactions with external stakeholders and senior company management be handled?
    - How can groups integrate people who are not co-located?
    - How can knowledge be shared across the group?



## Group Organization

- Small software engineering groups are usually organised informally without a rigid structure.
- For large projects, there may be a hierarchical structure where different groups are responsible for different sub-projects.
- Agile development is always based around an informal group on the principle that formal structure inhibits information exchange.



## Informal Groups

- The group acts as a whole and comes to a consensus on decisions affecting the system.
- The group leader serves as the external interface of the group but does not allocate specific work items.
- Rather, work is discussed by the group as a whole and tasks are allocated according to ability and experience.
- This approach is successful for groups where all members are experienced and competent.



# Working Environments

- Work place makes an important effect on worker's performance and their job satisfaction.
- The most important environmental factors identified:
  - **Privacy:** Programmers require an area where they can concentrate and work without interruption.
  - **Outside awareness:** People prefer to work in natural light and with a review of the outside environment.
  - **Personalization:** Individuals adopt different working practices and have different opinions on décor. The ability to rearrange the workplace to suit working practices and to personalize that environment is important.



## Summary

- Good project management is essential if software engineering projects are to be developed on schedule and within budget.
- Software management is distinct from other engineering management. Software is intangible. Projects may be novel or innovative with no body of experience to guide their management.
- Software processes are not as mature as traditional engineering processes.



## Summary

- Cost estimating is one of the most important steps in software project management, which are carried out together with project scheduling.
  - There is not necessarily a simple relationship between the price charged for a system and its development costs.
  - There are various techniques of software cost estimation. In preparing an estimate, several different techniques should be used. If the estimates diverge widely, this means that inadequate estimating information is available.
-



## Summary

- Risk management is now recognized as one of the most important project management tasks.
- Risk management involves identifying and assessing project risks to establish the probability that they will occur and the consequences for the project if that risk does arise.
- You should make plans to avoid, manage or deal with likely risks if or when they arise.



## Summary

- Software development groups should be fairly small and cohesive. The key factors that influence the effectiveness of a group are the people in that group, the way that it is organized and the communication between group members.
- Communications within a group are influenced by factors such as the status of group members, the size of the group, the gender composition of the group, personalities and available communication channels.