Software Engineering

HUANG Jie

School of Software Engineering

Tongji University, 2022

同济大学 TONGJI UNIVERSITY

Chapter 5
Human Aspects of Software Engineering

In chapter 3, we have learned

- ✓ Agile methods in software development process.
- ✓ Agile principles.
- ✓ Extreme programming(XP).
- ✓ Scrum.
- ✓ Kanban.

Chapter 5
Human Aspects of Software Engineering

In this chapter, we will discuss

- ✓ Characteristics of software engineer.
- ✓ Behavioral Model for Software Engineering.
- ✓ Organization and Team Structures.
- ✓ Effective Software Team Attributes.
- ✓ How to establish Team?
- ✓ Agile Team.
- ✓ Factors Affecting Global Software Development Team.

Traits of Successful Software Engineers

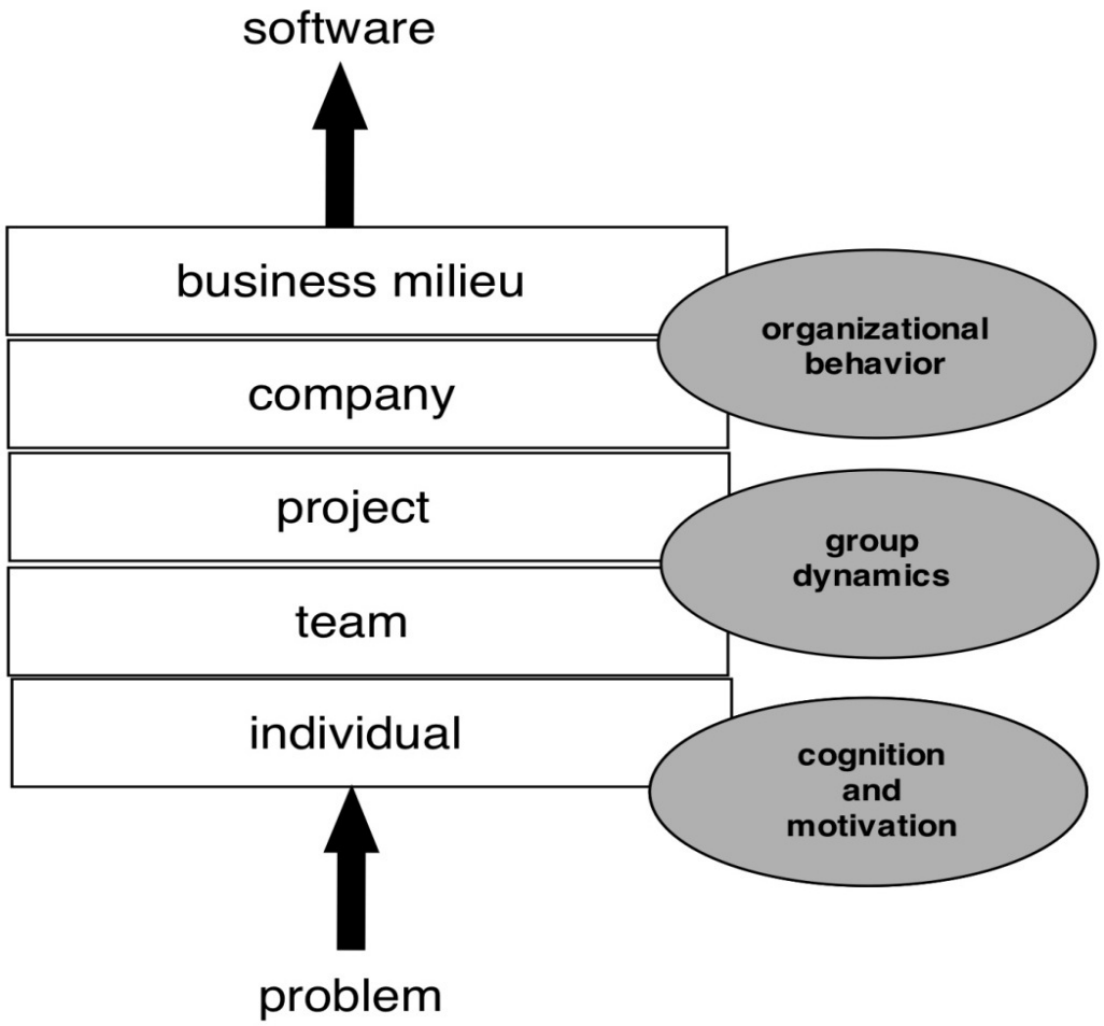
- Sense of individual responsibility.
- Acutely aware of the needs of team members and stakeholders.
- Brutally honest about design flaws and offers constructive criticism.
- Resilient under pressure.
- Heightened sense of fairness.
- Attention to detail.
- Pragmatic.

Boundary Spanning: Team Roles

- Ambassador – represents team to outside constituencies.
- Scout – crosses team boundaries to collect information.
- Guard – protects access to team work products.
- Sentry – controls information sent by stakeholders.
- Coordinator – communicates across the team and organization.
- **Q**uestion: Do you propose other roles in your team?
- **E**xample: The Surgical Team

From 《The Mythical Man-Month Essays on Software Engineering》 Chapter 3. by Frederick P.Brooks

Behavioral Model for Software Engineering



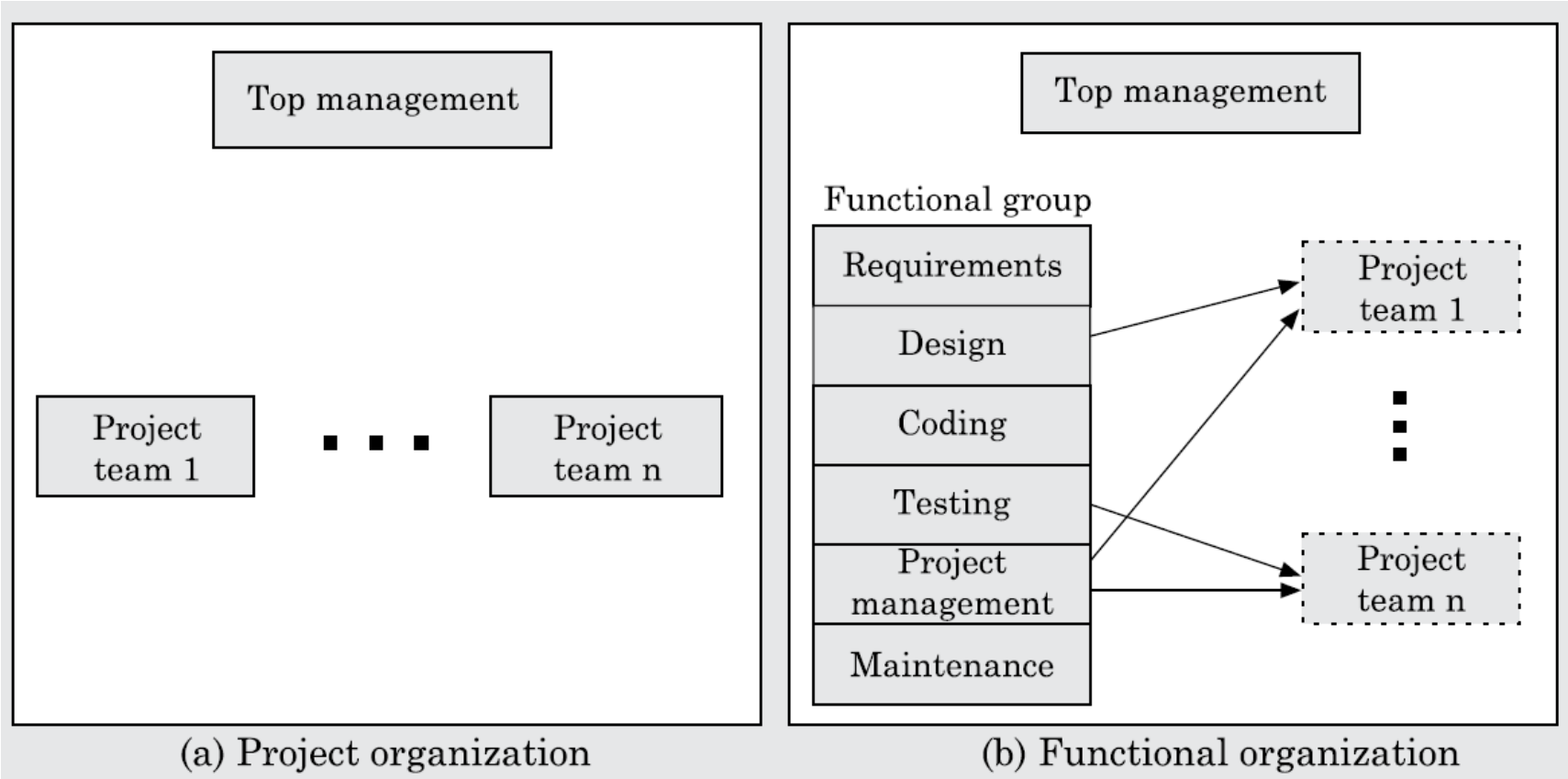
Organization & Team Structures

- A software development organization handles several projects at any time. With regard to organization, there are **two** important issues:
  - How is the **organization** as a whole structured (organization structure)?
  - How are the individual project **team**s structured (team structure)?
- Organization structure
  - Functional format.
  - Project format.
  - Matrix format.
- Team structure
  - Chief programmer team.
  - Democratic team.
  - Mixed control team organization.

Organization & Team Structures

- Organization structure – Functional format
  - The staff are divided based on the specific functional group to which they belong to. Different teams of programmers from different functional groups perform different phases of a project.
  - Feature: The functional format requires development of good quality documents because the work of one functional team must be clearly understood by the subsequent functional teams working on the project.
- Organization structure – Project format
  - The staff are divided based on the project for which they work. A set of developers is assigned to every project at the start of the project, remain with the project till the completion of the project.
  - Feature: The poor manpower utilization, since the full project team is formed since the start of the project, and there is very little work for the team during the initial phases of the life cycle.

Organization & Team Structures



(a) Project organization

(b) Functional organization

Organization & Team Structures

- Functional format vs Project format organization
  - The advantages of functional format
    - Efficient project staffing.
    - Production of good quality documents.
    - Job specialization.
    - Efficient handling of the problems associated with manpower turnover.
  - The disadvantage of project organization structure
    - Keep almost a constant number of developers for the entire duration of the project. Idling in the initial phase, tremendous pressure in the later phase of development.

Organization & Team Structures

- Why Functional format not be used popular in industry?
  - Considering the present skill shortage, it would be very difficult for the functional organizations to fill slots for some roles such as the maintenance, testing, and coding groups.
  - The functional organization structure is different to implement for small and medium sized organization, unless the company handles a large number of such projects.
  - For obvious reasons the functional format is not suitable for small organizations handling just one or two projects.

Organization & Team Structures

- Matrix format
  - This format is intended to provide the advantages of both functional and project structures.
  - The pool of functional specialists are assigned to different projects as needed. However, the functional teams report to both the project manager and the functional manager.
  - The deployment of the different functional specialists in different projects can be represented in a Matrix.
  - Observe the following figure, you can see the different members of a functional specialists are assigned to different projects.

Organization & Team Structures

- Matrix format
  - In a matrix organization, project manager needs to share responsibilities for the project with a number of individual functional managers.

	Project			
Functional group	#1	#2	#3	
#1	2	0	3	Functional manager 1
#2	0	5	3	Functional manager 2
#3	0	4	2	Functional manager 3
#4	1	4	0	Functional manager 4
#5	0	4	6	Functional manager 5
	Project manager 1	Project manager 2	Project manager 3	

Organization & Team Structures

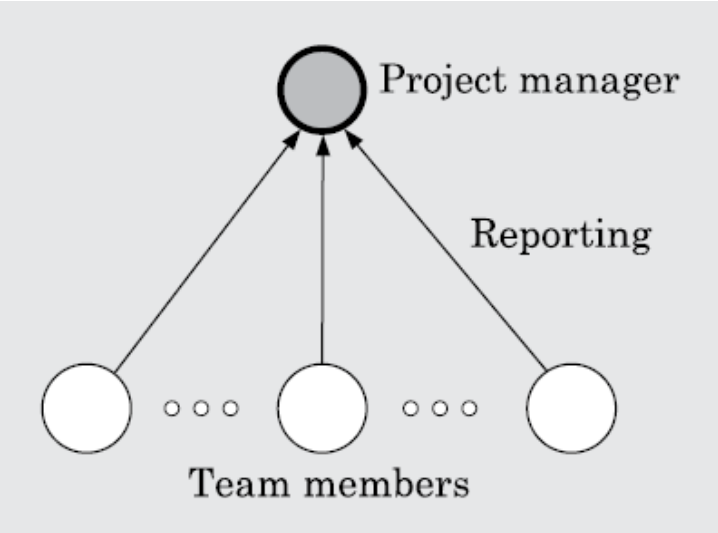
- Matrix format organization
  - Matrix organizations can be characterized as **weak** or **strong**, depending on the relative authority of the functional managers and project managers.
    - In a strong functional matrix, the functional managers have authority to assign workers to projects and project managers have to accept the assigned personnel.
    - In a weak matrix, the project manager controls the project budget, can reject workers from functional groups, or even decide to hire outside workers.
  - Two problems of matrix format organization
    - Due to the multiplicity of authority, conflicts can occur between functional and project managers over allocation of workers.
    - In a strong matrix organization, frequent shifting of workers may take place as the functional managers adopt a firefighting mode to tackle the crises in different projects.

Organization & Team Structures

- Team structure
  - Projects of specific complexities and sizes often require specific team structures for efficient working. Generally, three kinds of team structure are used in practice.
    - Chief programmer team.
    - Democratic team.
    - Mixed control team.

Organization & Team Structures

- Team structure - Chief programmer team
  - This organization is probably the most efficient way of completing simple and small projects since the chief programmer can quickly work out a satisfactory design and then ask the programmers to code different modules of his design solution.
  - Suppose an organization has successfully completed many simple MIS projects. Then, for a similar MIS project, chief programmer team structure can be adopted. The chief programmer team structure works well when the task is within the intellectual grasp of a single individual.

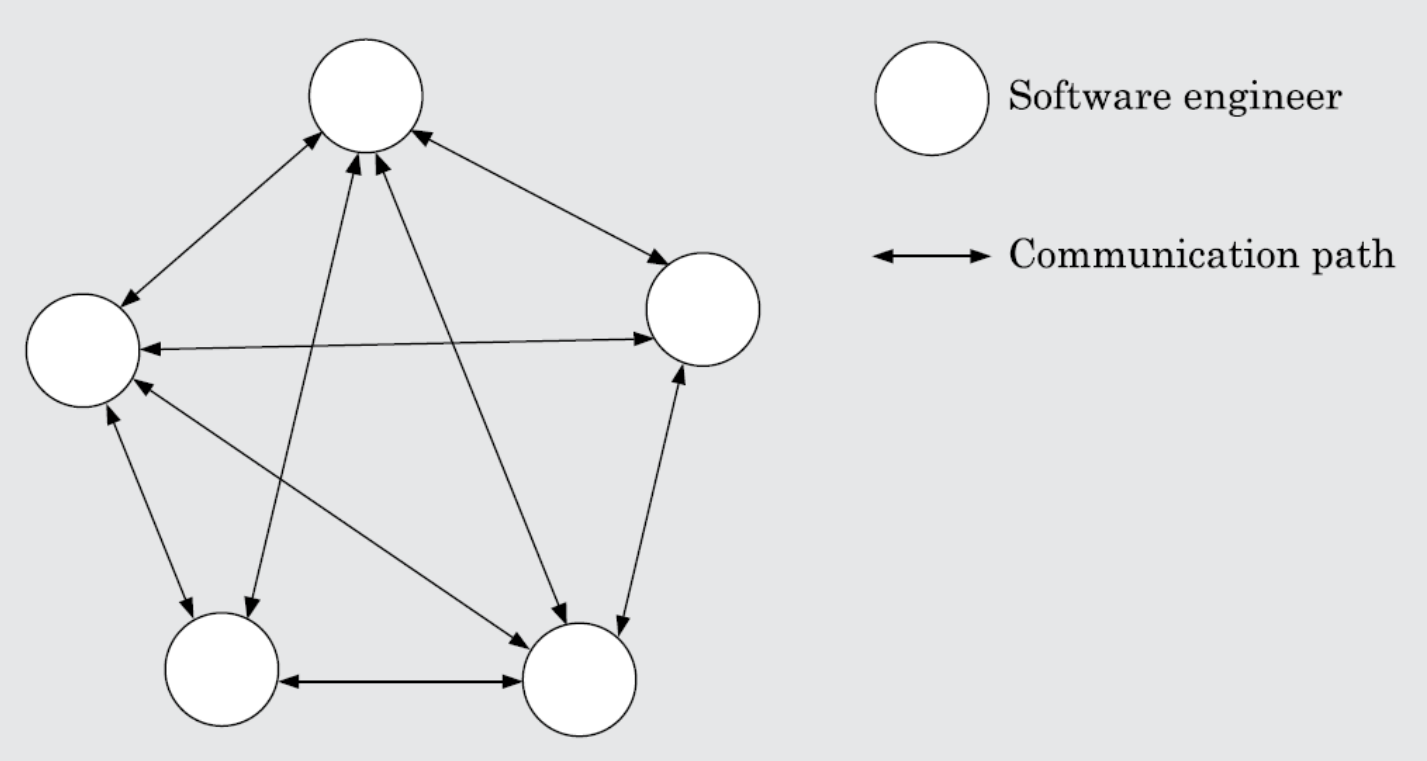


Organization & Team Structures

- Team structure – Democratic team
  - Typically, a manager provides the administrative leadership. At different times, different members of the group provide technical leadership.
  - In a democratic organization, the team members have higher morale and job satisfaction. Consequently, it suffers from less manpower turnover.
  - Though the democratic teams are less productive compared to the chief programmer team, the democratic team structure is appropriate for less understood problems, since a group of developers can invent better solutions than a single individual as in a chief programmer team.
  - A democratic team structure is suitable for research-oriented projects requiring less than five or six developers. For large sized projects, a pure democratic organization tends to become chaotic. The democratic team organization encourages *egoless programming* as programmers can share and review each other's work.
  - Disadvantage: the team members may waste a lot time arguing about trivial points due to the lack of any authority in the team to resolve the debates.

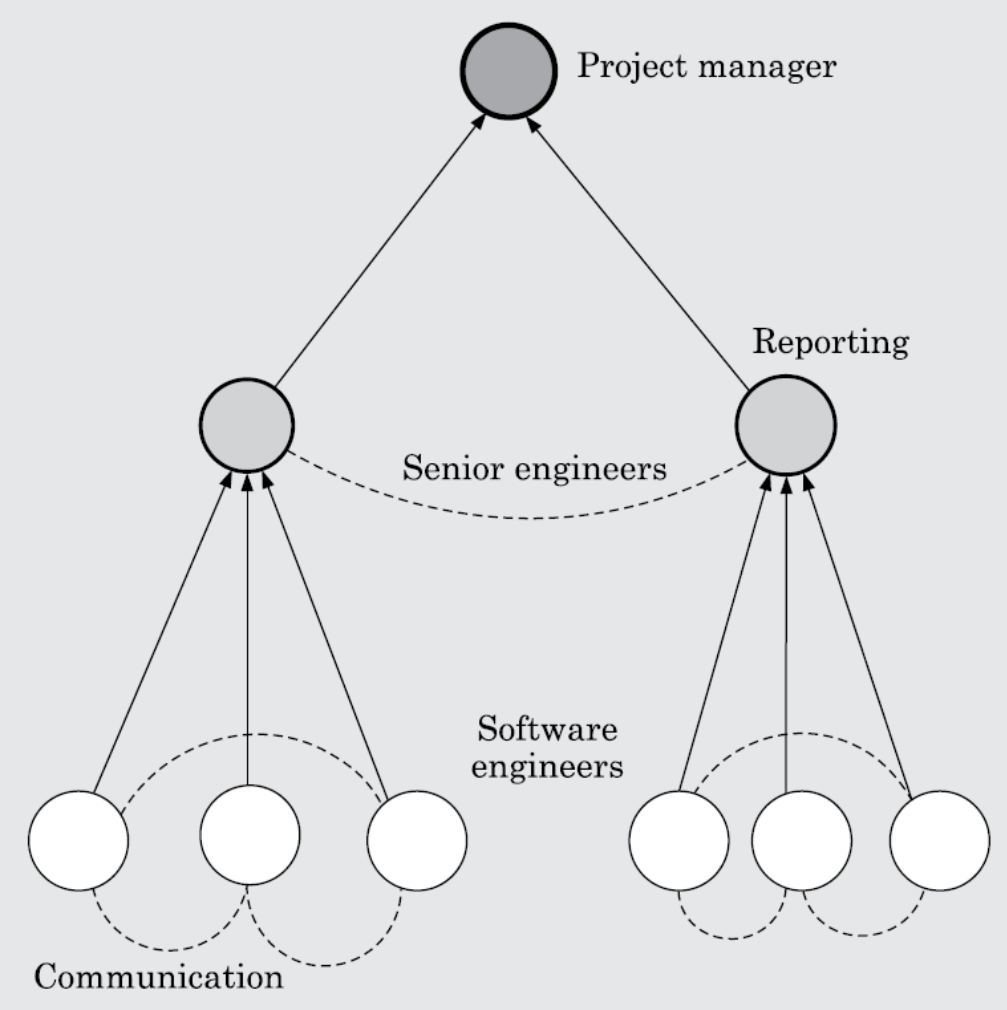
Organization & Team Structures

- Team structure – Democratic team



Organization & Team Structures

- Team structure – Mixed control team



Organization & Team Structures

- Team structure – Mixed control team
  - This kind of team organization draws upon the ideas from both the democratic organization and the chief-programmer organization.
  - This team organization incorporates both hierarchical reporting and democratic set up.
  - In above figure, the communication paths are shown as dashed lines and the reporting structure is shown using solid arrows.
  - The mixed control team organization is suitable for large team sizes. The democratic arrangement at the senior developers level is used to decompose the problem into small parts.
  - Each democratic setup at the programmer level attempts solution to a single part. Thus, this team organization is eminently suited to handle large and complex programs. This team structure is extremely popular and is being used in many software development companies.

Factors Affecting Team Structure

- The following factors must be considered when selecting a software project team structure ...
  - The difficulty of the problem to be solved.
  - The size of the resultant program(s) in lines of code or function points.
  - The time that the team will stay together (team lifetime).
  - The degree to which the problem can be modularized.
  - The required quality and reliability of the system to be built.
  - The rigidity of the delivery date.
  - The degree of sociability (communication) required for the project.

Effective Software Team Attributes

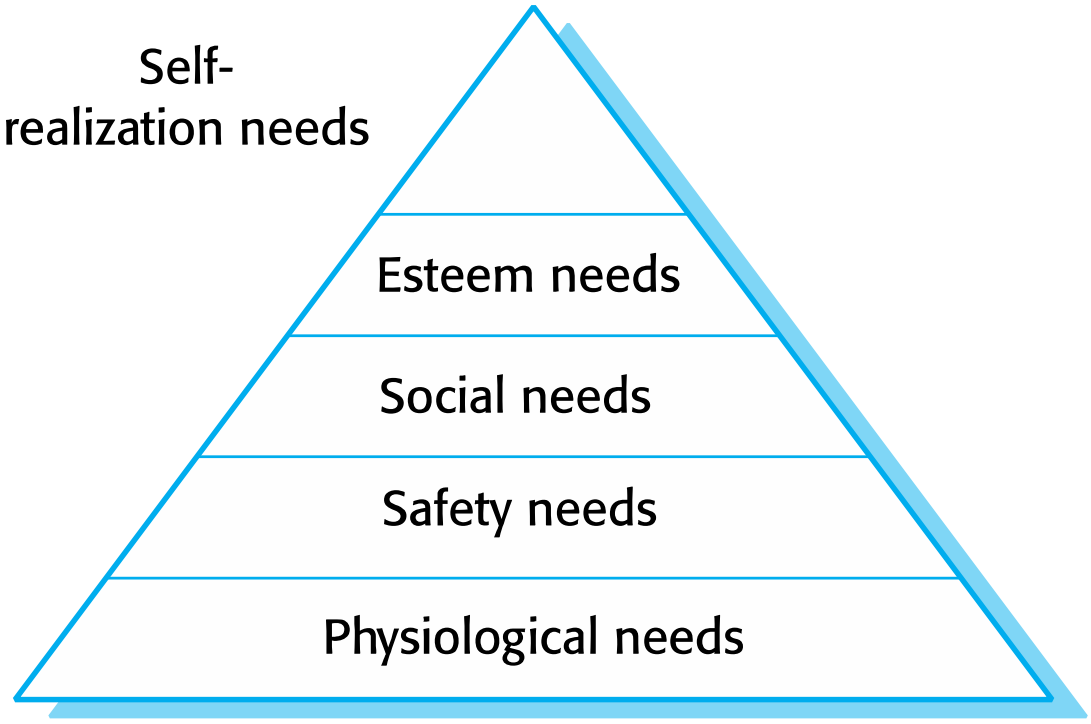
- Sense of purpose.
- Sense of involvement.
- Sense of trust.
- Sense of sustainable improvement.
- Diversity of team member skill sets.
- ...

Avoid Team "Toxicity"

- A frenzied work atmosphere in which team members waste energy and lose focus on the objectives of the work to be performed.
- High frustration caused by personal, business, or technological factors that cause friction among team members.
- "Fragmented or poorly coordinated procedures" or a poorly defined or improperly chosen process model that becomes a roadblock to accomplishment.
- Unclear definition of roles resulting in a lack of accountability and resultant finger-pointing.
- "Continuous and repeated exposure to failure" that leads to a loss of confidence and a lowering of morale.

Human needs hierarchy

- 5 Layer Structure – Maslow,1968



Who is a good software engineer?

- The following attributes that good software developers should possess:
  - Exposure to systematic techniques, i.e. familiarity with software engineering principles.
  - Good technical knowledge of the project areas(*Domain knowledge*).
  - Good programming abilities.
  - Good communication skills. These skills comprise of oral, written, and interpersonal skills.
  - High motivation.
  - Sound knowledge of fundamentals of computer science.
  - Intelligence.
  - Ability to work in a team.
  - Discipline, etc.

Who is a good software engineer?

- Since software development is a group activity, it is vital for a software developer to possess three main kinds of communication skills - **Oral, Written,** and **Interpersonal**.
- A software developer not only needs to effectively communicate with his teammates (e.g., reviews, walkthroughs, and other team communications) but may also have to communicate with the customer to gather product requirements.
- A software developer is also expected to document his work (design, code, test, etc.) as well as write the users' manual, training manual, installation manual, maintenance manual, etc. This requires good ***written communication*** skill.
- **Motivation level** of a software developer is another crucial factor contributing to his work quality and productivity.

CASE STUDY: INDIVIDUAL MOTIVATION

Alice is a software project manager working in a company that develops alarm systems. This company wishes to enter the growing market of assistive technology to help elderly and disabled people live independently. Alice has been asked to lead a team of 6 developers than can develop new products based around the company's alarm technology.

Alice's assistive technology project starts well. Good working relationships develop within the team and creative new ideas are developed. The team decides to develop a peer-to-peer messaging system using digital televisions linked to the alarm network for communications. However, some months into the project, Alice notices that Dorothy, a hardware design expert, starts coming into work late, the quality of her work deteriorates and, increasingly, that she does not appear to be communicating with other members of the team.

CASE STUDY: INDIVIDUAL MOTIVATION

Alice talks about the problem informally with other team members to try to find out if Dorothy's personal circumstances have changed, and if this might be affecting her work. They don't know of anything, so Alice decides to talk with Dorothy to try to understand the problem.

After some initial denials that there is a problem, Dorothy admits that she has lost interest in the job. She expected that she would be able to develop and use her hardware interfacing skills. However, because of the product direction that has been chosen, she has little opportunity for this. Basically, she is working as a C programmer with other team members.

Although she admits that the work is challenging, she is concerned that she is not developing her interfacing skills. She is worried that finding a job that involves hardware interfacing will be difficult after this project. Because she does not want to upset the team by revealing that she is thinking about the next project, she has decided that it is best to minimize conversation with them.

How to establish SE team?

- A number of project factors that should be considered when planning the structure of software engineering(SE) teams [Mantei81]

  (1) Difficulty of the problem to be solved.

  (2) Size of the resultant programs in line of code or FP.

  (3) Time that the team will stay together (team lifetime).

  (4) Degree to which the problem can be modularized.

  (5) Required quality and reliability of the system to be built.

  (6) Rigidity of the delivery date.

  (7) Degree of sociability(communication) required for project.

Organizational Paradigms

- Closed paradigm

  structures a team along a traditional hierarchy of authority.

- Random paradigm

  structures a team loosely and depends on individual initiative of the team members .

- Open paradigm

  attempts to structure a team in a manner that achieves some of the controls associated with the closed paradigm but also much of the innovation that occurs when using the random paradigm.

- Synchronous paradigm

  relies on the natural compartmentalization of a problem and organizes team members to work on pieces of the problem with little active communication among themselves.

Generic Agile Teams

- Stress individual competency coupled with group collaboration as critical success factors.
- People trump process and politics can trump people.
- Agile teams as self-organizing and have many structures
  - An adaptive team structure.
  - Uses elements of Constantine's random, open, and synchronous structures.
  - Significant autonomy.
- Planning is kept to a minimum and constrained only by business requirements and organizational standards.

XP Team Values

- Communication – close informal verbal communication among team members and stakeholders and establishing meaning for metaphors as part of continuous feedback.
- Simplicity – design for immediate needs nor future needs.
- Feedback – derives from the implemented software, the customer, and other team members.
- Courage – the discipline to resist pressure to design for unspecified future requirements.
- Respect – among team members and stakeholders.

TEAM SPIRIT

Alice, an experienced project manager, understands the importance of creating a cohesive group. As they are developing a new product, she takes the opportunity of involving all group members in the product specification and design by getting them to discuss possible technology with elderly members of their families. She also encourages them to bring these family members to meet other members of the development group.

Alice also arranges monthly lunches for everyone in the group. These lunches are an opportunity for all team members to meet informally, talk around issues of concern, and get to know each other. At the lunch, Alice tells the group what she knows about organizational news, policies, strategies, and so forth. Each team member then briefly summarizes what they have been doing and the group discusses a general topic, such as new product ideas from elderly relatives.

Every few months, Alice organizes an ‘away day’ for the group where the team spends two days on ‘technology updating’. Each team member prepares an update on a relevant technology and presents it to the group. This is an off-site meeting in a good hotel and plenty of time is scheduled for discussion and social interaction.

Impact of Social Media

- Blogs – can be used share information with team members and customers.
- Microblogs (e.g. Twitter) – allow posting of real-time messages to individuals following the poster.
- Targeted on-line forums – allow participants to post questions or opinions and collect answers.
- Social networking sites (e.g. Wechat, Facebook, LinkedIn) – allows connections among software developers for the purpose of sharing information.
- The distinct benefits of social media must be weighed against the treat of uncontrolled disclosure of private information.

Software Engineering using the Cloud

- Benefits
  - Provides access to all software engineering work products.
  - Removes device dependencies and available everywhere.
  - Provides avenues for distributing and testing software.
  - Allows software engineering information developed by one member to be available to all team members.

- Concerns
  - Dispersing cloud services outside the control of the software team may present reliability and security risks.
  - Potential for interoperability problems becomes high with large number of services distributed on the cloud.
  - Cloud services stress usability and performance which often conflicts with security, privacy, and reliability.

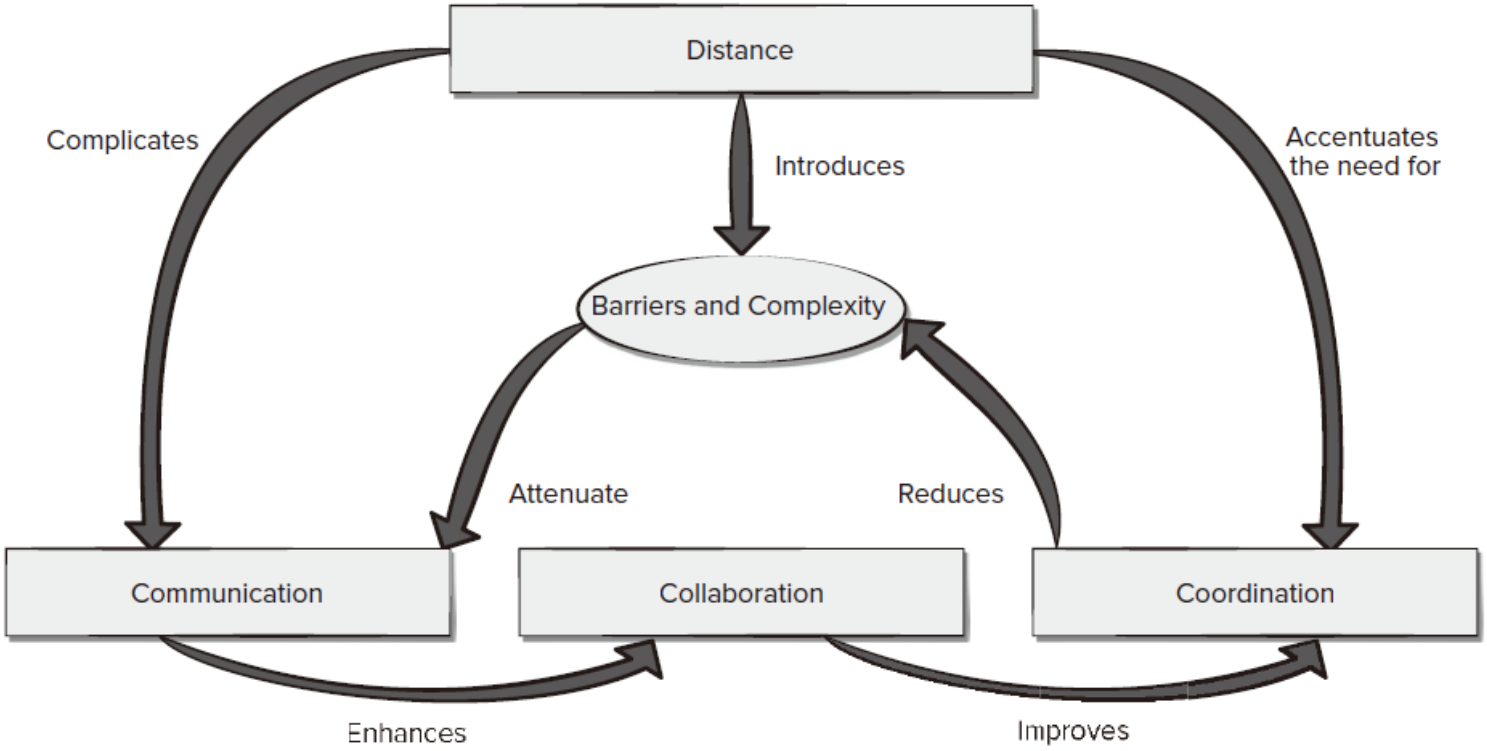
Collaboration Tools

- Namespace that allows secure, private storage or work products.
- Calendar for coordinating project events.
- Templates that allow team members to create artifacts that have common look and feel.
- Metrics support to allow quantitative assessment of each team member's contributions.
- Communication analysis to track messages and isolates patterns that may imply issues to resolve.
- Artifact clustering showing work product dependencies.
- Question: *How to use GitHub as a collaboration tool in your team? e.g. branch or fork?*

Team Decisions Making Complications

- Problem complexity.
- Uncertainty and risk associated with the decision.
- Work associated with decision has unintended effect on another project object (law of unintended consequences).
- Different views of the problem lead to different conclusions about the way forward.
- Global software teams face additional challenges associated with collaboration, coordination, and coordination difficulties.

Factors Affecting Global Software Development Team



Summary

- A successful software engineer must have technical skills. But in addition, he must take responsibility for his commitments, be aware of the needs of his peers, be honest in his assessment of the product and the project, be resilient under pressure, treat his peers fairly, and exhibit attention to detail.
- A successful software team is more productive and motivated than average. To be effective, a software team must have a sense of purpose, a sense of involvement, a sense of trust, and a sense of improvement. In addition the team must avoid toxicity that is characterized by a frenzied and frustrating work atmosphere, an inappropriate process, an unclear definition of roles on the software team, and continuous exposure to failure.

Assignment

- Chapter 5: Problems to ponder

5.1 Based on your personal observation of people who are excellent software developers, name three personality traits that appear to be common among them.

5.5 Referring to Figure 5.2, why does distance complicate communication? Why does distance accentuate the need for coordination? What types of barriers and complexities are introduced by distance?

Assignment

- Reading

  《The Mythical Man-Month: Essays on Software Engineering》

  by Frederick P.Brooks

  Chapter 7 Why Did the Tower of Babel Fail?

- Preview

  《Software Engineering》(8th Edition)

  by R.S.Pressman

  Chapter 6 Principles that guide practice

huangjie@tongji.edu.cn