2019 级本科

软件工程

Software Engineering

张昕

zhangxin@cust.edu.cn

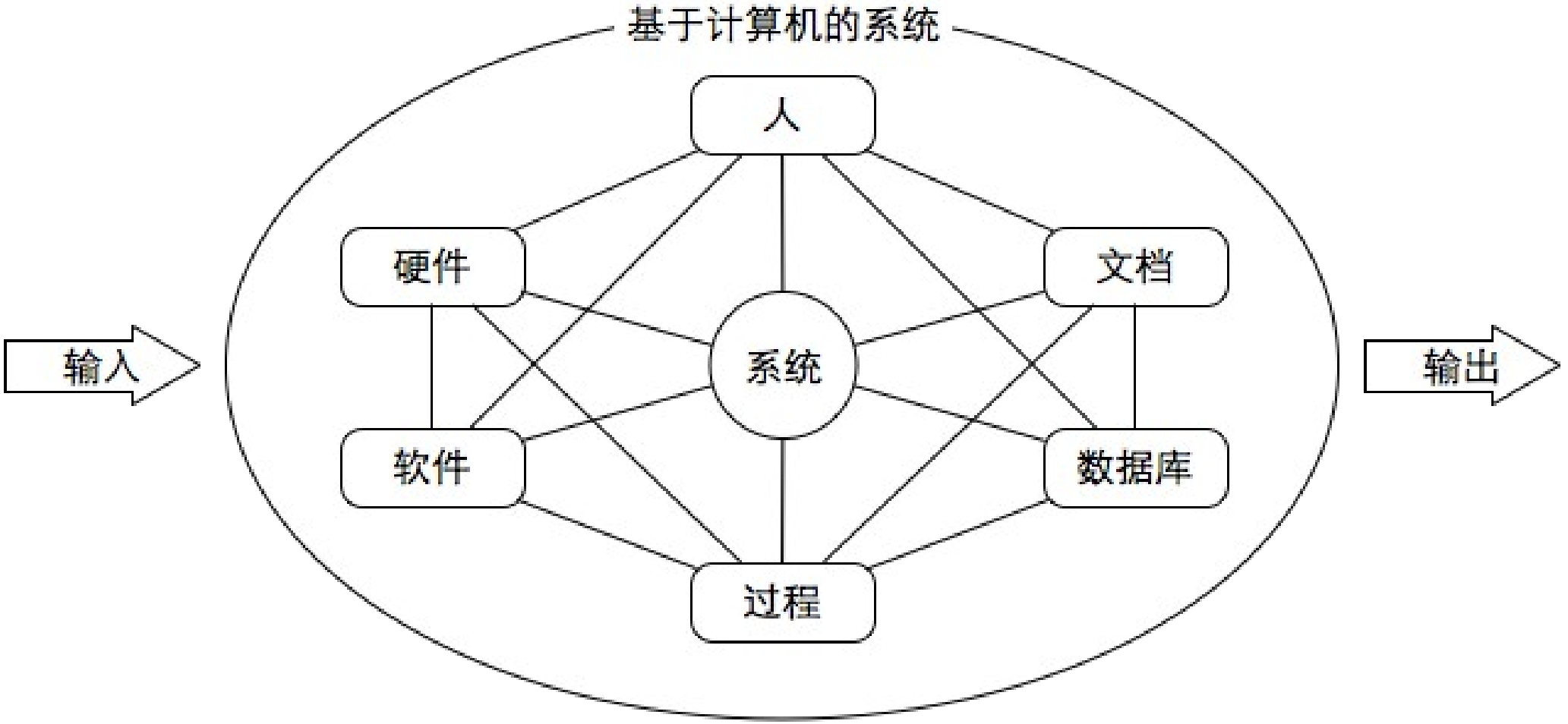
计算机科学技术学院软件工程系

Software planning

Computer system

- Generally, a computer-based system is composed of system elements such as hardware, software, people, documents, databases, and processes.
  - Without considering the internal structure and functions of the system, a computer-based system can be represented by an input-processing-output (IPO) model.
    - I (Input) refers to the input of information;
    - P (Process) refers to the processing of information;
    - O (Output) refers to the output of information.
  - For a large-scale computer-based system, the element itself may also be a computer-based system. The system referred to here will have a complex hierarchical structure.

Computer-based system



Role of software

- Software plays an important role in all parts of the IPO model of a computer-based system
  - Realize the input and output of the system
    - The input information of the system comes from external entities of the system, and the input of a certain subsystem inside the system may come from other subsystems outside or inside the system
    - The software can provide an interactive man-machine interface to realize the logic functions of human-machine interaction such as prompt mechanism, data input and output, and use the software driver to drive and control the hardware to complete the input and output operations
  - If necessary, the software can set up an interface with the database to support the system's access to the database
  - The software makes various system elements work together through a series of algorithms and operation control programs, so as to realize the functions and performance of the system

Software in-house lifecycle

- Tasks of the software development part
  - Software is to transform the system's requirements for software into operational system elements,
  - Development of software consists of three stages
    - Overall design
    - Process design
    - Coding
- Software overall design stage
  - Refers to the overall software structure design and data design, the main tasks at this stage
    - Design the modular structure of the software
    - Define the interface and establish the data structure
    - Generate outline design specifications and assembly test plans
    - Review the quality of the outline design, focusing on whether the overall design supports the completeness and traceability of the software requirements specification

Software in-house lifecycle

- Software process design phase
  - Main mission
  - A detailed description of the process of each module in the outline design specification
  - Develop a unit test plan and generate detailed design specifications
  - Review the stage product of the detailed design
- Coding stage
  - main mission
    - Use the selected programming language to convert the detailed process description of each module into a program
      - Attention should be paid to good programming style, simplicity and self-documentation, while maintaining traceability to process design

Software Verification, Acceptance, Maintanance

- The main tasks of the software verification phase
  - Software developers perform unit tests on each module according to the unit test plan
    - Verify that the function of the module is correct and meets the design requirements
  - Organize developers and dedicated software test engineers to conduct comprehensive tests on the software
    - Test whether the overall structure and interface of the software meet the design requirements, and test whether each software component meets the corresponding software functional requirements and performance requirements
  - Organize experts, users and customers to review the test results
- The main tasks of software submission and distribution
  - Develop a formal user manual, classify, organize, and archive documents, and establish a configuration control mechanism
  - Submit the software to the user and, if necessary, be responsible for installing the software into the user's environment

Software Verification, Acceptance, Maintanance

- The main tasks of software maintenance
  - Correct the errors found during the operation of the software
  - Improve the functionality and performance of the software
  - Adapt to changes in the software operating environment
  - Improve the maintainability and reliability of the software, etc.

Feasibility analysis and study

- Motivation for feasibility study
  - The development of any computer-based system will be limited by time and resources
  - Before accepting the customer's project, the developer must conduct a feasibility study based on the time and resources that the customer may provide to ensure that the project is accessible, usable, and usable
- Tasks of the feasibility study
  - After preliminary requirements definition
  - Determine whether the project is worth solving in the shortest time with the least cost and whether there is a feasible solution
    - That is to demonstrate the feasibility of system development at the system level
  - It is necessary to analyze the pros and cons of several main possible solutions to determine whether the original system scale and goals are realistic, and whether the benefits after the completion of the system are large enough to be worth the investment in the development of this system.

NOT think about how to solve the problem

Feasibility analysis and study – category

- Economic feasibility study
  - Estimate the development cost of the project and the possible profit after it is put into use
  - Conduct cost-benefit analysis and impact on other products or profits
- Technical feasibility study
  - According to the system function, performance requirements and various constraints of the realization of the system proposed by the customer, the feasibility of the realization of the system is studied from a technical point of view
- Operation and operation feasibility study
  - The main research is whether the operation mode of the system can be effectively implemented in the user unit and whether it is inconsistent with other original systems
  - Whether the operating procedures of the system are feasible in the user's unit
    - Including personnel, technology policies, management methods, etc.

Feasibility analysis and study – category [cont.]

- Legal feasibility study
  - Study whether the development and use of the new system will infringe the rights and interests of others, and whether it violates the laws and regulations of the country
- Choice of development plan
  - The main task of the feasibility study is to make recommendations for future actions
    - If there is no feasible solution to the problem, the analyst should recommend stopping the project to avoid further waste
    - If the problem is worth solving, put forward and evaluate various feasible development schemes for realizing the system, choose one of the best schemes, and make a preliminary development plan for the system

Steps of the feasibility study

- Review the preliminary analysis results
  - Review the preliminary analysis results and reports of the system, correct vague or inaccurate statements, re-determine the system goals and scale, and clearly describe all constraints on the system
  - The purpose is to ensure that the problem that the analyst is solving is indeed the problem that he is required to solve.
  - The most fundamental task of the feasibility study is to make recommendations for the future course of action.
    - If there is no feasible solution to the problem, the analyst should recommend stopping the development project to avoid wasting time, resources, manpower and money;
    - If the problem is worth solving, the analyst should recommend a better solution and develop a preliminary plan for the project.
  - Feasibility study The length of time required for a feasibility study depends on the scale of the project. Generally speaking, the cost of a feasibility study is only 5% to 10% of the expected total cost of the project.

Steps of the feasibility study

- Research existing systems
  - The existing system is an important source of information. The new target system must also be able to complete its basic functions;
  - If the existing system is perfect, users will naturally not ask for the development of a new system. It can be seen that the existing system must have some shortcomings, and the new system must be able to solve the problems in the old system.
  - Find out its basic functions and information, point out its shortcomings or limitations
    - The cost of running and using the old system is an important economic indicator. If the new system cannot increase revenue or reduce usage costs, the new system is not as good as the old system from an economic point of view.
- A common mistake is to spend too much time analyzing the existing system.
    - The purpose of "researching existing systems" is to understand what the existing system can do, not how it does these tasks. The analyst should draw a high-level system flow chart depicting the existing system, and ask the person concerned to check whether his understanding of the existing system is correct. Do not spend too much time to understand and describe the implementation details of the existing system.
  - No system operates in a "vacuum", and most systems are connected to other systems.
    - Should pay attention to understand and record the interface between the existing system and other systems, which is an important constraint when designing a new system.

Overall Objectives and Philosophy

Analysis Rules of Thumb

- The model should focus on requirements that are visible within the problem or business domain. The level of abstraction should be relatively high. "Don't get bogged down in details" that try to explain how the system will work.
- Each element of the requirements model should add to an overall understanding of software requirements and provide insight into the information domain, function, and behavior of the system.
- Delay consideration of infrastructure and other nonfunctional models until design. That is, a database may be required, but the classes necessary to implement it, the functions required to access it, and the behavior that will be exhibited as it is used should be considered only after problem domain analysis has been completed.

Overall Objectives and Philosophy

Analysis Rules of Thumb [cont.]

- Minimize coupling throughout the system. It is important to represent relationships between classes and functions. However, if the level of "interconnectedness" is extremely high, efforts should be made to reduce it.
- Be certain that the requirements model provides value to all stakeholders. Each constituency has its own use for the model. For example, business stakeholders should use the model to validate requirements; designers should use the model as a basis for design; QA people should use the model to help plan acceptance tests.
- Keep the model as simple as it can be. Don't add additional diagrams when they add no new information. Don't use complex notational forms when a simple list will do.

Requirements engineering

- Domain Analysis

- The broad spectrum of tasks and techniques that lead to an understanding of requirements is called requirements engineering.
  - From a software process perspective, requirements engineering is a major software engineering action that begins during the communication activity and continues into the modeling activity.
  - It must be adapted to the needs of the process, the project, the product, and the people doing the work.
  - Requirements engineering builds a bridge to design and construction.