

Topic 4: 系统开发与低代码/零代码平台

System Development and Low-code/No-code Development Platform (LCDP)

刘跃文 博士 Dr. LIU, Yuewen

教授、博士生导师 Professor

liuyuewen@xjtu.edu.cn

西安交通大学管理学院

School of Management, Xi'an Jiaotong University

V2.0, 2023-Oct

提纲 Outline

1. 典型的信息系统结构
A General Network Structure of Information System
2. 传统的系统开发方法
Traditional System Development Method
3. 低代码/零代码开发
Low-code/No-code Development Platform (LCDP)
4. 低代码开发的发展方向
Future Directions of LCDP
5. 作业 I
Homework I

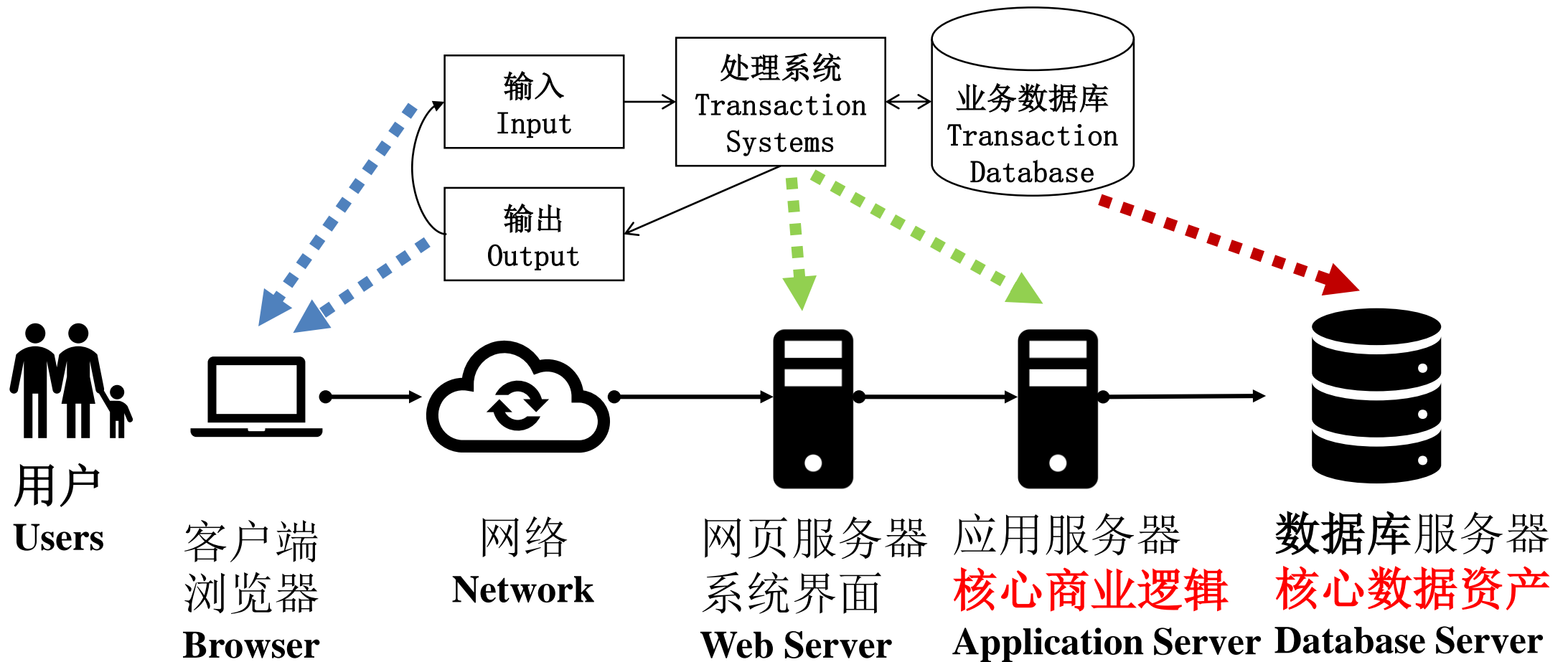
1. 信息系统网络结构

A General Network Structure of Information System



1. 信息系统网络结构

The Network Structure of an Information System



1. Enter the keyword “**管理信息系统教材**” and send a page access request
`https://search.jd.com/Search?keyword=管理信息系统&enc=utf-8&pvid=****`

2. Preset basic web page framework without content
`https://search.jd.com/Search`, request to fill in multiple specific contents on the webpage, such as product lists, recommended product lists, etc

3. Based on the preset data acquisition/calculation method, send operation requests to the database, such as querying products in the product table that include certain keyword

4. Execute the query and provide results, such as `select * from goods where name like '%管理信息系统%'`

Data sheet

7. Finds results such as “**管理信息系统教材**” products and recommended products

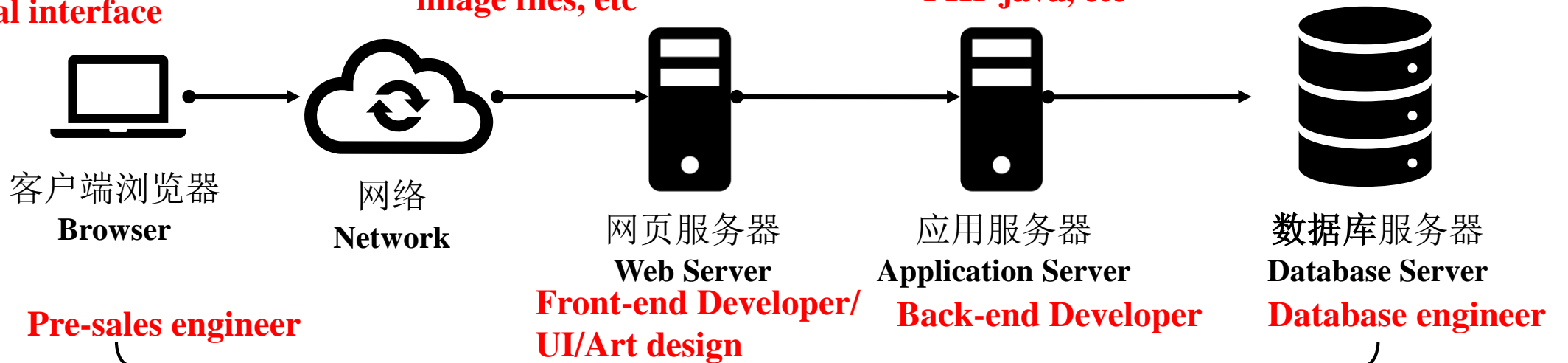
6. Combine into webpage content

5. Calculate and process the query results (such as sorting) to form content that meets the requirements of the webpage

Graphical interface

Html CSS js image files, etc

PHP java, etc



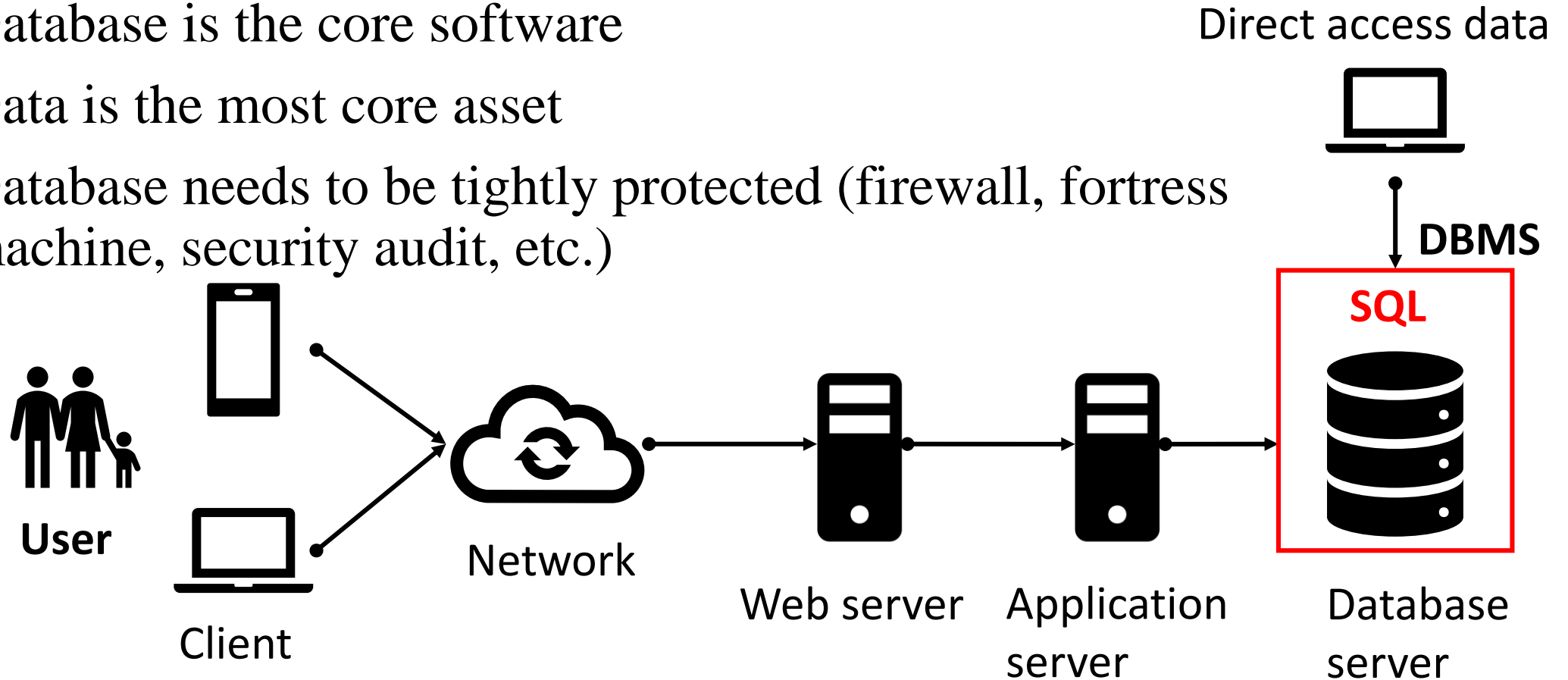
2. 三种服务器

Three Server Types

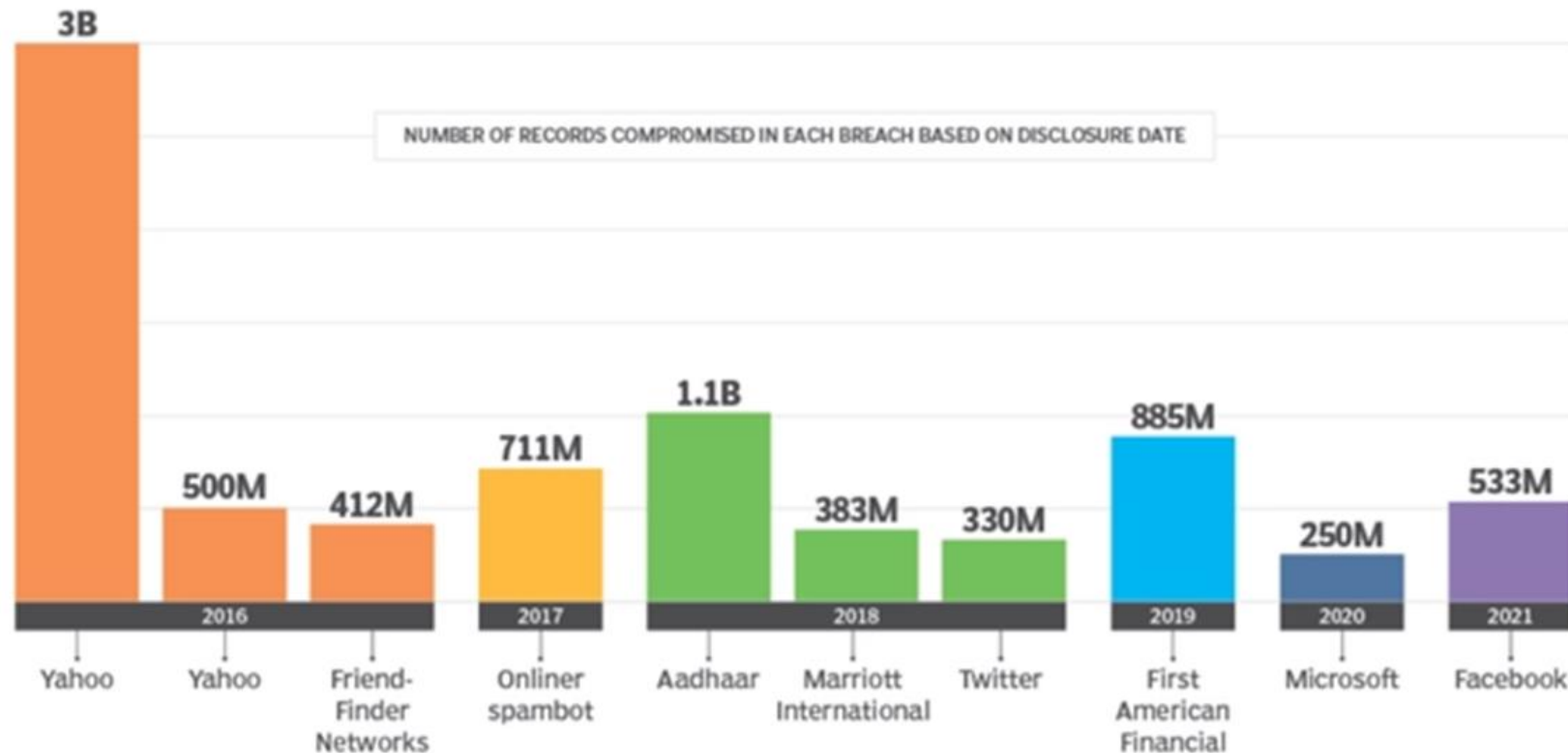
- Web服务器-用户界面Web Server - User interface
- 应用服务器-商业逻辑Application Server-Business logic
 - 一些商业逻辑非常简单（例如：查询）
Some business logic is very simple (such as querying)
 - 一些数据计算方法非常复杂（例如：排程、调度、推荐、优化）
Some data calculation methods are very complex (e.g., scheduling, scheduling, recommendation, optimization)
- 数据库服务器-数据资产Database server - Data asset
 - 一些数据库非常简单Some databases are very simple
 - 一些数据库非常复杂（完整性约束、依赖关系、权限控制、操作的原子性、并发控制等） Some databases are very complex (integrity constraints, dependencies, permission control, atomicity of operations, concurrency control, etc.)

So why can't we see the database?

- Database is the core software
- Data is the most core asset
- Database needs to be tightly protected (firewall, fortress machine, security audit, etc.)



The consequences of being "dragged" are very serious



Source: Top Ten Data Discovery Incidents in History <https://baijiahao.baidu.com/s?id=1740121852776165027>

Can we use files (such as Excel) to store data? [understand]

- 7 questions to store data with files:
- Data isolation — multiple files and formats
 - Data storage is too decentralized, data islands, too many formats
- Data redundancy and inconsistency
 - Repeated information is stored in different files in different data formats
- Difficulty in accessing data
 - For different needs, you need to write different access procedures
- Integrity problems
 - Integrity constraints (such as bank balance > 0) needs to be written in the program code instead of being given by the outside; it is difficult to increase or modify the integrity constraint

Can we use files (such as Excel) to store data? [understand]

- Atomicity of updates
 - If only a part of an operation is completed, the consistency of the data may be destroyed
 - For example: the movement of money from one account to another account, either completes completely, does not happen at all, and cannot be executed to half
 - For example: What will happen to delete data to half of the power off?
- Concurrent access by multiple users
 - Concurrent access requirements often appears, and uncontrolled concurrent access may cause data inconsistencies
 - For example: Two people take 100 yuan each from one account at the same time, but the account balance is only 100
- Security problems
 - It's hard to provide someone with part of the data

2. 传统的系统开发方法

Traditional System Development Method

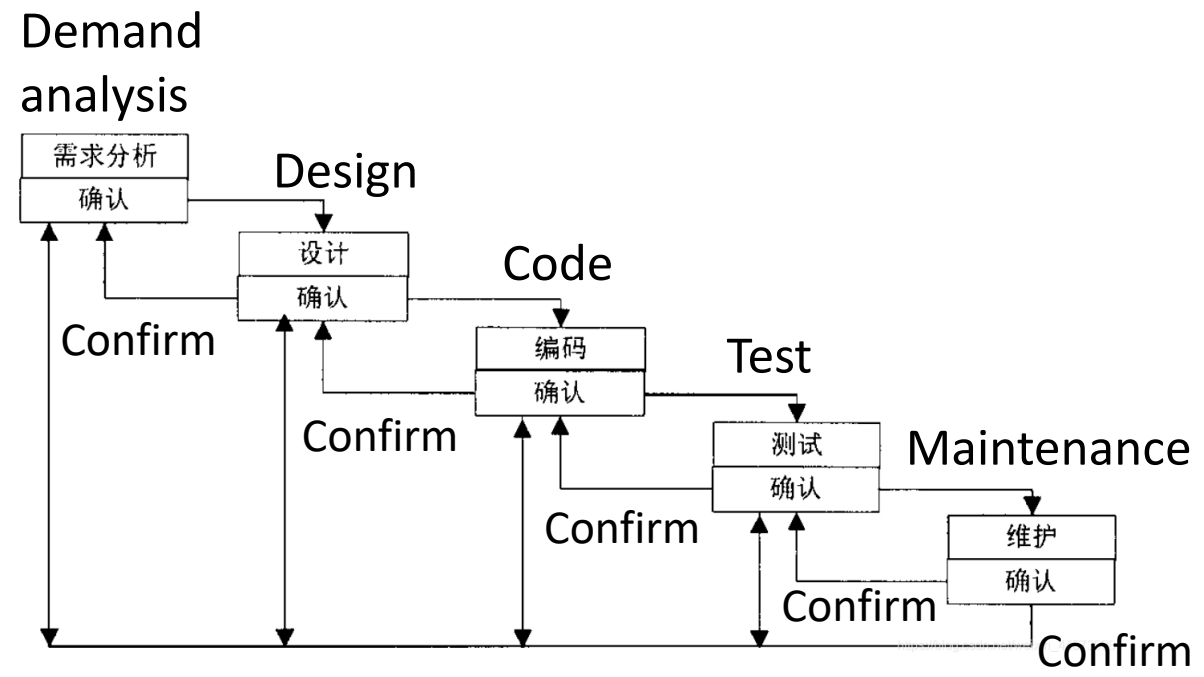
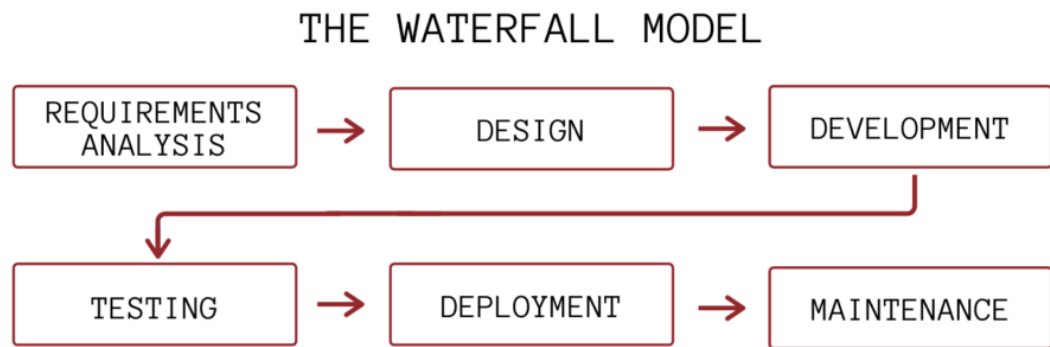


Whose job is it to build/develop a system?

- CTO/CIO/Leader: It's the job of the IT department
- IT department: I pay for it, so it is the job of software/system development companies.
- Common process:
 - Writing feasibility study report, budget
 - Expert review (feasibility study report)
 - Compile bidding instructions, bidding: pay start-up funds
 - Preliminary design-Expert review (Preliminary design)
 - Detailed design- Expert review (Detailed design)
 - Code development and testing
 - Acceptance and payment

1. Waterfall Model

- Waterfall model (also known as software survival cycle model or linear sequential process model) was first proposed by W.Royce in 1970, providing a systematic and sequential method of software development.
- It is a model of several stages of the software survival cycle as a linear sequence, including problem definition, feasibility study, demand analysis, summary design, detailed design, coding, testing, and maintenance.
- Waterfall model starts from demand analysis and gradually proceeds until the software products that are confirmed by the user. The output result of the previous stage of the waterfall model is the input of the next stage, just like the waterfall flowing water, and the falling is down step by step.
- Waterfall model provides a template that allows the method of analysis, design, coding, testing, and maintenance to have a common guidance in this template.



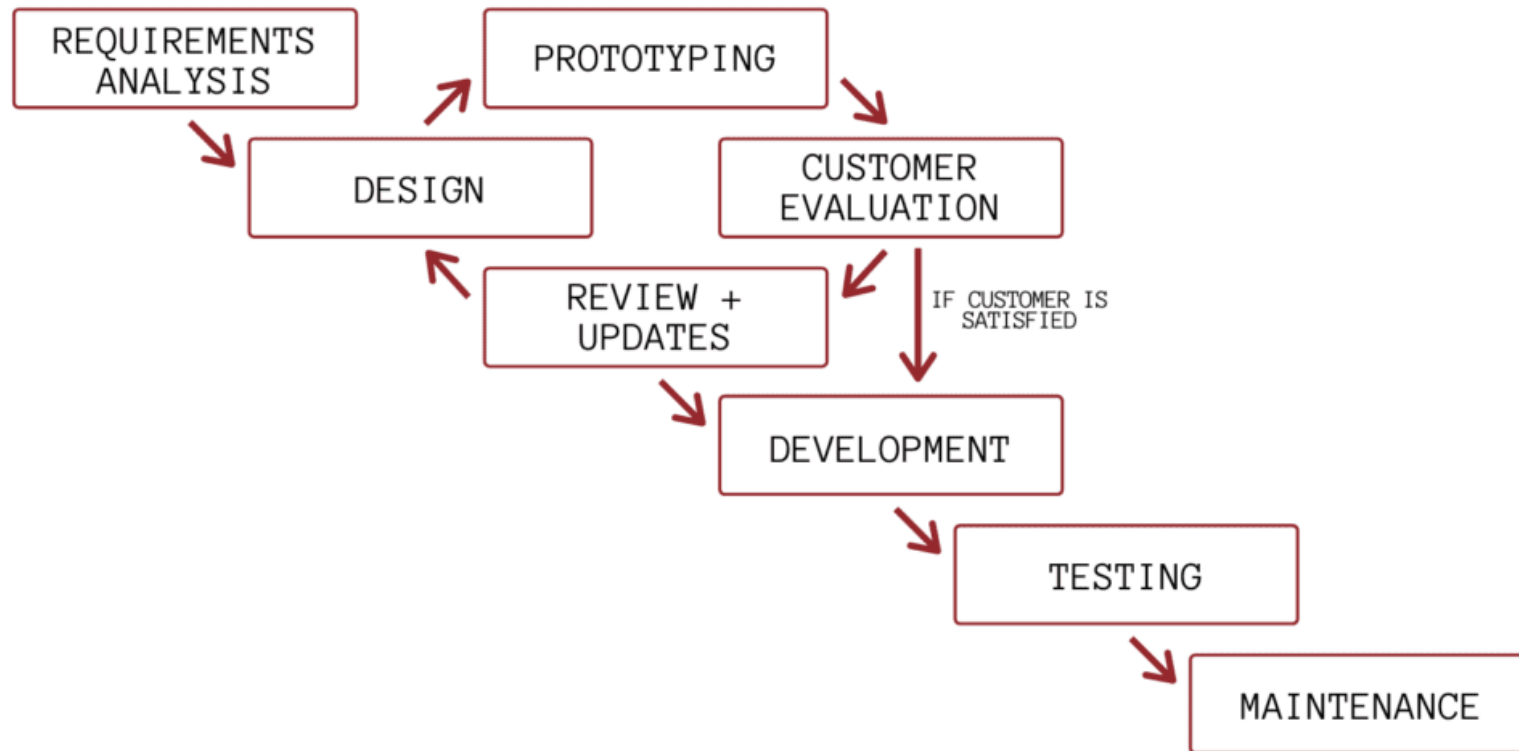
A confused start + A disastrous end

- In most cases, it is difficult to follow the order given by the model in actual projects, and the iteration of this model is indirect, which can easily cause significant confusion due to small changes.
- Users often cannot accurately express their needs, but this model requires it. This model does not welcome ambiguity issues.
- Users have to wait until the late stages of the development cycle to see the test version. At this point, if a big mistake is found, it may cause panic, and the consequences may be catastrophic.
- Using this linear model, it is often necessary to wait for other members to complete the process at the beginning and end before proceeding, which may take longer than the development time. We call it 'blockage state'.

2. Prototyping Model

- Prototyping model is a model that can gradually improve into a running system. Based on a preliminary understanding of user requirements, developers rely on their own understanding of user needs, supported by powerful software environments and utilizing rapid software development tools, to build, design, and develop a tangible initial software model (prototype, an implementable software application model).
- Developers and users have reached an agreement on the "prototype". In this way, it can reduce the errors in design and risks in development, and also reduce the time for user training, thereby improving the practicality, correctness and user satisfaction of the system.
- Prototyping model adopts a gradual refinement method to improve the prototype, enabling rapid development and avoiding the lengthy development process like Waterfall model.
- Prototyping model continues to improve through "samples" to reduce costs.

PROTOTYPING MODEL

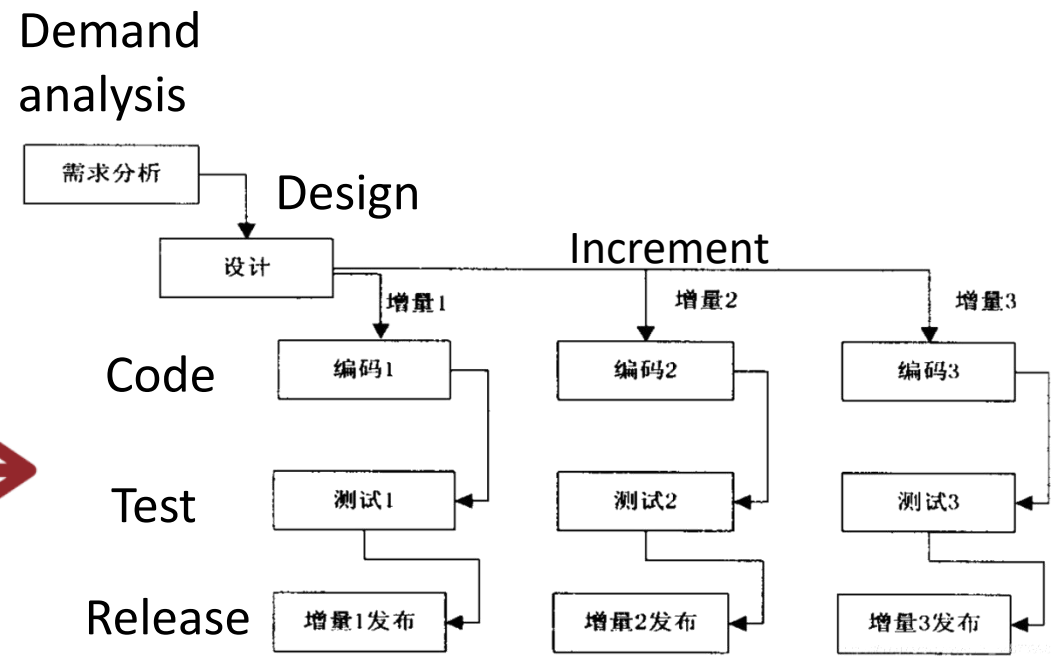
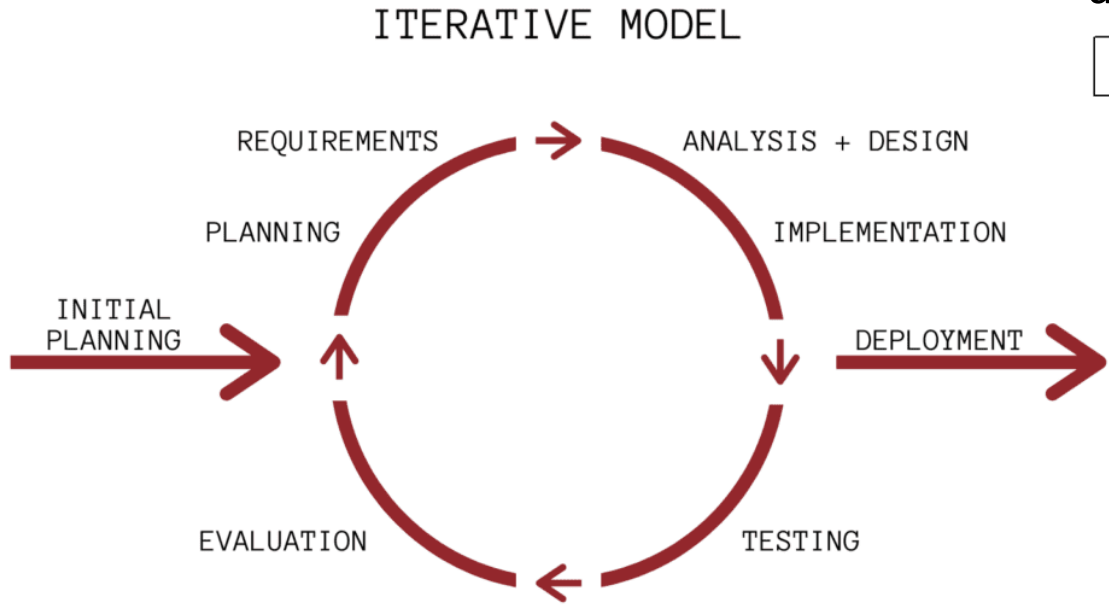


Prototyping Model was destined to be abandoned

- Users see a running version of the software, but is unaware that the prototype is temporary. Software developers do not consider the overall quality or future maintainability of the software in order to make the prototype run as soon as possible. When told that the product must be rebuilt in order to achieve high quality, users often cry foul.
- Developers often need to compromise on implementation to get prototypes working as quickly as possible. Developers are more likely to adopt an inappropriate operating system or programming language just because it is generic or well-known, or to use an inefficient algorithm just for demonstration purposes. After a while, developers may become so used to these choices that they forget why they are inappropriate. These undesirable choices then become part of the software.
- When developing a system using Prototyping Model, users and developers must reach an agreement on the following issues: the prototype is only built for user to define requirements, and should not be used as a final product. It is necessary to fully consider aspects such as quality and maintainability before starting the final software development.

3. Iterative (and Incremental) Model

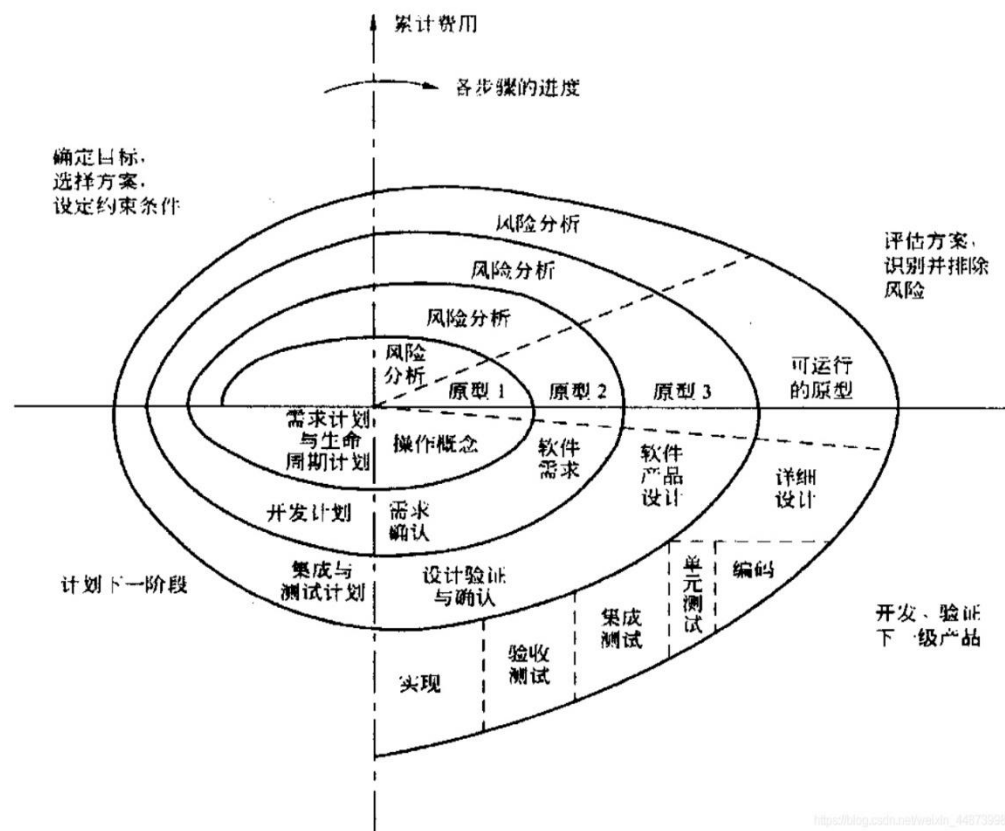
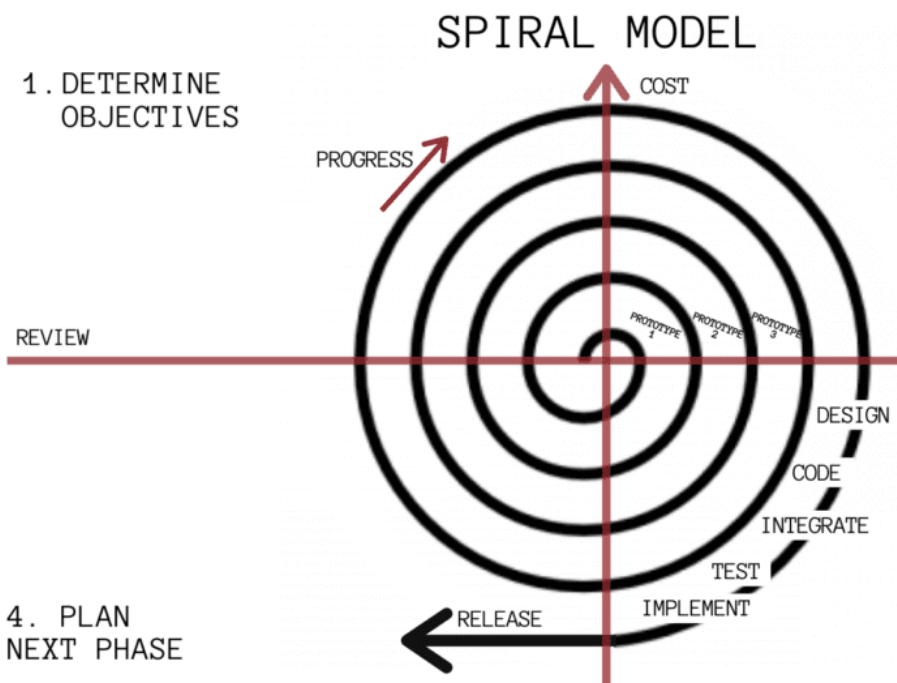
- Iterative model (also known as incremental model) combine the basic components of linear sequential models with the iterative features of prototype implementation models. Iterative model uses linear sequences that interlace as the schedule time progresses. Each linear sequence produces a releasable "increment" of the software.
- When using the incremental model, the first increment is often the core product, that is, the first increment implements the basic requirements, but many additional features have not yet been released. Each increment is used and evaluated by users as new features and functionality released in the next increment. This process is repeated after each incremental release until the final, complete product is produced. The incremental model emphasizes that each increment releases an operational product.



-
- During the iterative process, some development resources are wasted.
 - Indefinable resource consumption and indefinable deadlines.
 - From beginning to end, developers and users are entangled until the full version comes out.

4. Spiral Model

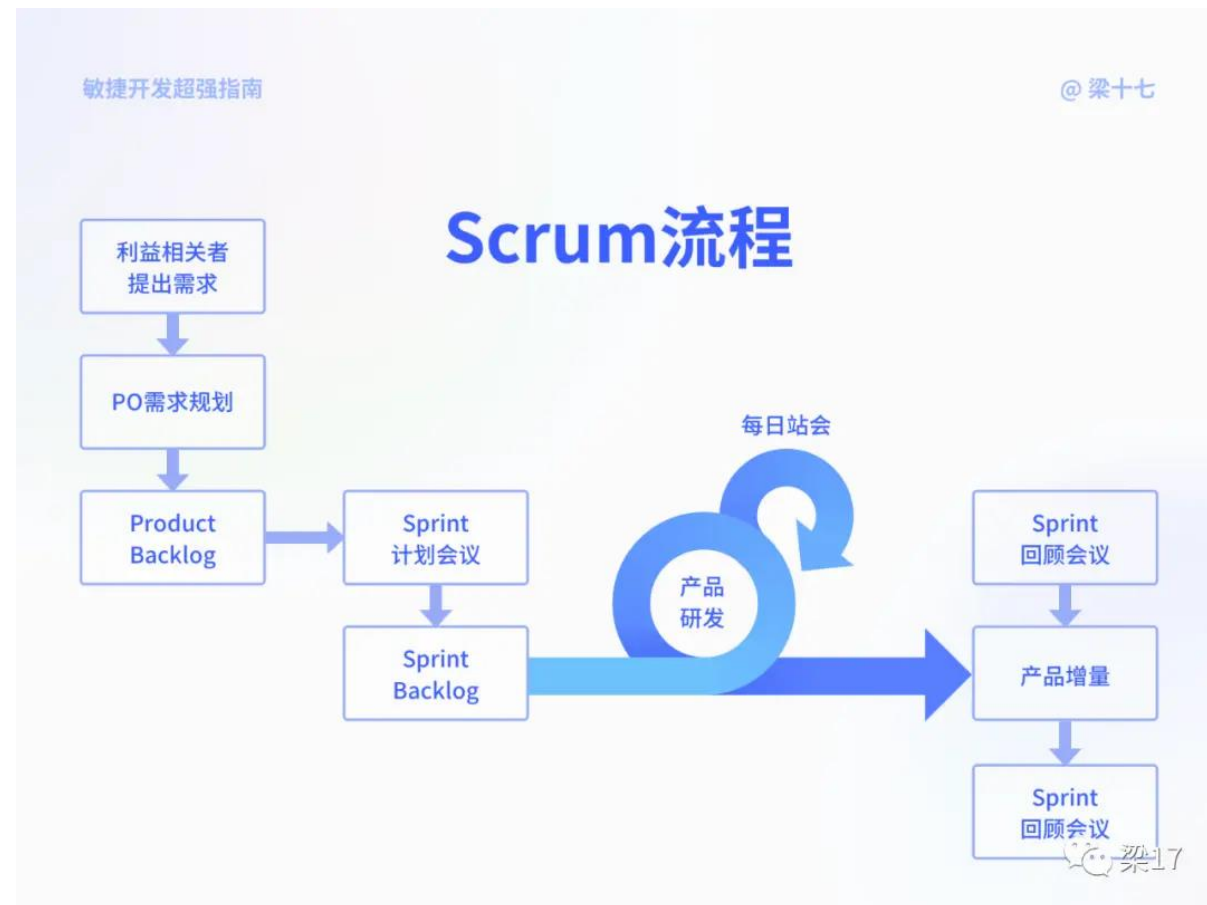
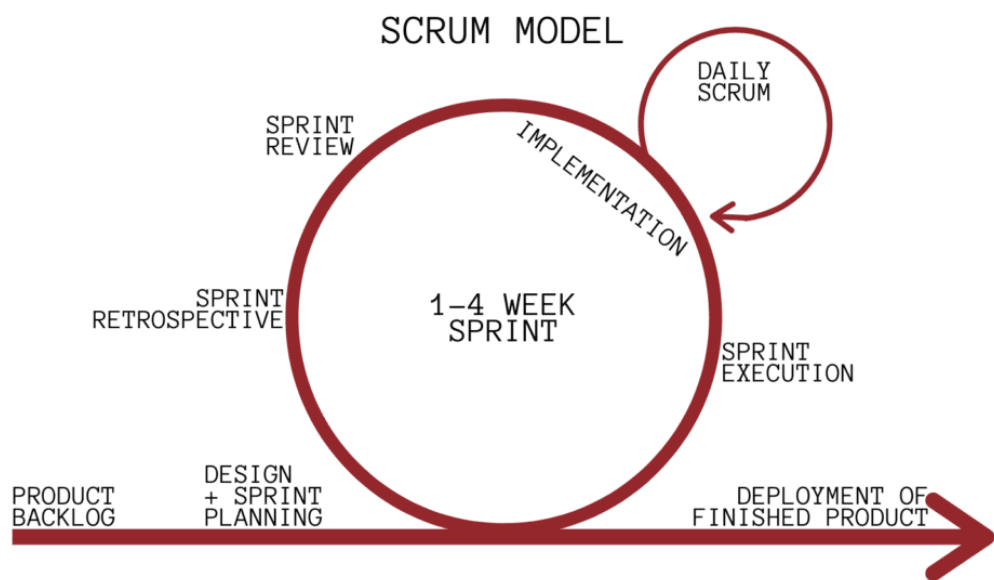
- Spiral model is an evolutionary software process model that combines the iterative characteristics of prototype implementation with the controlled and systematic aspects of a linear sequential model to enable the rapid development of incremental versions of software.
- In Spiral model, software development is a series of incremental releases. With each iteration, a more complete version of the system being developed is gradually produced. The spiral model is divided into several framework activities, also known as task areas. For large software, developing only one prototype is often not enough. The spiral model combines the waterfall model and the incremental model, and adds risk analysis.
- Spiral model divides the development process into several spiral cycles, each of which can be divided into four work steps: 1 Determine goals, plans, and constraints. 2. Evaluate the plan, identify risks, and address risks. 3. Develop and confirm products. 4. Plan for the next cycle of work.



-
- Techniques for risk analysis and assessment are required, and success depends on these techniques.
 - Obviously, if there is a big risk that is not detected, there will be problems, and it may even cause the evolution process to lose control. This model is relatively new and not widely used, and its efficacy needs further verification.

5. Agile Software Development

- Agile software development is an umbrella term for a set of methods and practices based on the values defined in the Agile Software Development Manifesto and Principles The Twelve Principles of Agile Software. Self-organizing, cross-functional teams evolve solutions using practices that are appropriate for their own environment. In other words, agile development is a kind of software development ability to cope with rapidly changing needs. As long as the development team can have the ability to cope with rapidly changing needs on the basis of values and principles, it is called agile development.
- For example: Scrum Agile development model
 - The whole process of Scrum is divided into 6 steps: 1. Product Backlog 2. Sprint Backlog 3. Sprint planning meetings 4. Sprint 5. Sprint assess meetings 6. Sprint review meetings



6. Revlon ERP project analysis

- In 2019, there was a class action lawsuit against Revlon, which was indirectly caused by the high cost of ERP secondary development. At that time, Revlon, as a well-known beauty product, chose one of the world's top ERP manufacturers. However, as the ERP project progressed, Revlon's demands continued to change, and ERP manufacturers' quotations became increasingly expensive. Eventually, overwhelmed, Revlon decided to suspend the project, which directly caused delays in project progress and resulted in the delayed delivery of \$64 million in products, leading to a class action lawsuit.
- What system development model might Revlon use for ERP development?
- What problems may arise from adopting a budget-development approach to project management?
- Is there a "best" system development model?

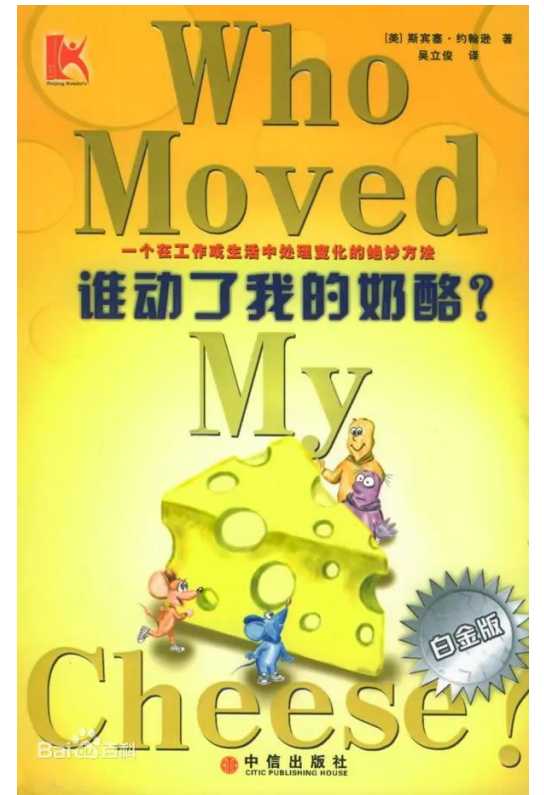
3. 低代码/零代码开发

Low-code/No-code Development Platform (LCDP)



1. Change is the only constant thing.

- Who Moved My Cheese
- The old adage asserts that, ‘change is the only constant thing’; consequently, adapting and enjoying change repeatedly is the best thing to do. Spencer wrote Who Moved My Cheese to illustrate the effects of denying change or accepting it.
- Smell The Cheese Often So You Know When It Is Getting Old.



刻舟求剑

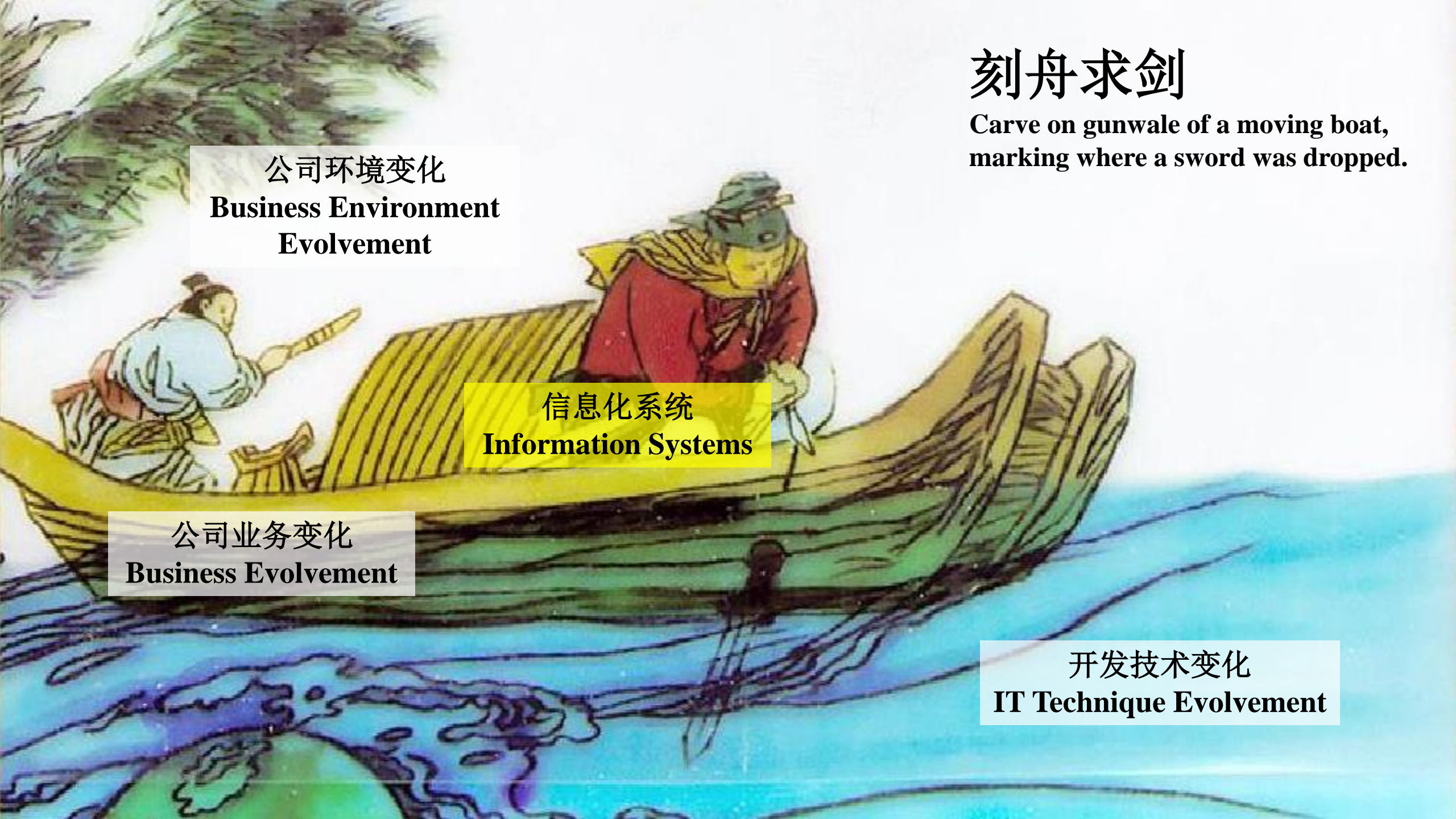
Carve on gunwale of a moving boat,
marking where a sword was dropped.

公司环境变化
Business Environment
Evolvement

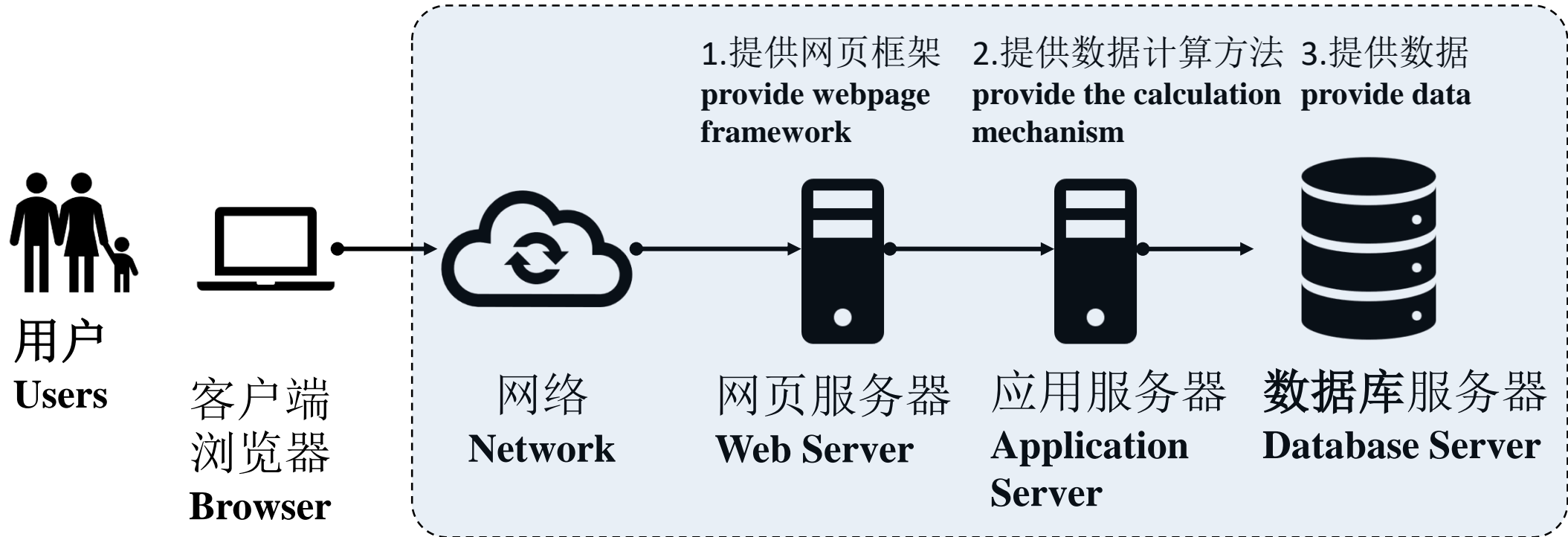
信息化系统
Information Systems

公司业务变化
Business Evolvment

开发技术变化
IT Technique Evolvment



云服务 Could Service



1. Encapsulate technical elements and quickly build applications from a business perspective.

2. Develop apps quickly with little or no code.

2.Low-code/No-code Development Platform

- Low-code is a technique and tool for rapidly developing systems with little or no code and for quickly configuring and deploying them. (From a concept proposed by Gartner in 2014).
- Low-code is a set of digital technology tool platform, based on graphical drag and drop, parametric configuration and other more efficient ways to achieve rapid construction, data orchestration, connection ecology, middle service. Apply innovation to scenarios in digital transformation with little or no code.
- Low-code platforms are also often referred to as aPaaS platforms. aPaaS, Application Platform as a Service. Apaas, application platform as a service. Gartner, an internationally renowned consulting firm, defines aPaaS as "a solution based on PaaS (Platform as a service) that supports the development, deployment and operation of applications in the cloud, and provides users with basic tools in software development, including data objects, rights management, and user interfaces." Users can quickly build, deploy, run and manage applications directly on the aPaaS platform in a Low-code/No-code manner.

← 正在编辑表单：客户公司

编辑字段 表单设置 公开发布 关闭

常用控件

- A 文本
- 数值
- 金额
- 日期
- 电话
- 多选
- 成员
- 附件
- 地区
- 定位

高级控件

- fx 公式
- 检查框
- ★ 等级
- 文本组合
- 并 自动编号
- 富文本

表单设计 14/200 快速排列

公司信息

- A 公司名称 1
- 座机
- A 网站
- 员工人数
- 主要联系人

单选

邮箱

请填写手机号码

请填写邮箱地址

请填写文本内容

A 地址

请填写文本内容

请填写数值

订单总收入

+ 记录

邮箱

字段名称

邮箱

默认值

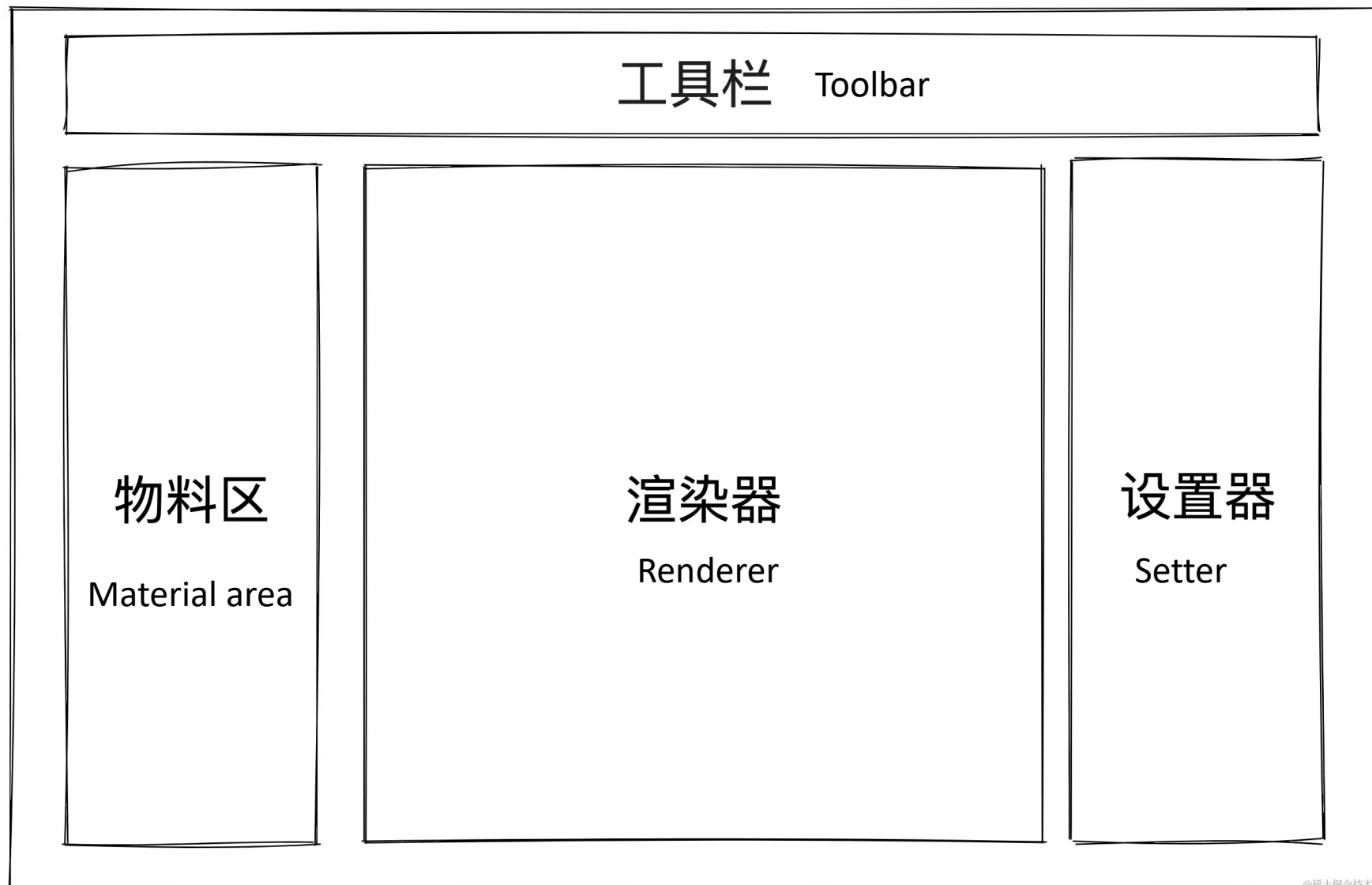
验证

- 必填
- 不允许重复

字段属性

- 只读
- 隐藏
- 新增记录时隐藏





© 清华大学出版社





B站视频6种低代码平台

先进生产力工具系列



钉钉宜搭简单视频教程



钉钉宜搭审批流程

3. LCDP Advantages

- Reduce development costs and operational barriers
 - Low code development tools abstract, unify, develop, and encapsulate functions to form modules that users can directly use, saving a lot of repetitive development labor.
 - Enable users who do not know how to program to utilize the platform's ready-made features and build the applications they need.
 - Business personnel can also use a zero code development platform to completely eliminate the operational barriers of digital applications.
- Improve the flexibility of the system to adapt to business changes
 - Build highly flexible business management applications that can modify configurations according to business changes at any time.
 - For both business and technical departments, low code can effectively improve the efficiency of problem-solving and reduce communication costs for both parties.
- Provide inspiration and tools for business model innovation
 - Low code development platforms endow non developers with the tools and spirit of daring to imagine and trial and error, which is precisely the driving force required to drive industry innovation and enterprise progress.

4. The Weaknesses of LCDP

- There are still learning and hands-on costs.
- Limited by components and features of low code platforms.
- The underlying code is unknown, and the card bug requires the cooperation of the platform company to solve or avoid.
- The number of concurrency and connections is limited by platform capabilities, making performance optimization difficult.
- The issue of upgrading and development.
 - The front-end framework is constantly changing, and after one or two years of reform, low code platforms need to be adapted once, which incurs significant maintenance costs;
 - The platform itself has been upgraded, and you don't know what impact it will have on your project, but without following the upgrade, you won't be able to iterate;
 - To switch from low code on one platform to other platforms, you need to go from 0 to 1.



一汽大众低代码应用

四川成都
一汽-大众成都分公司

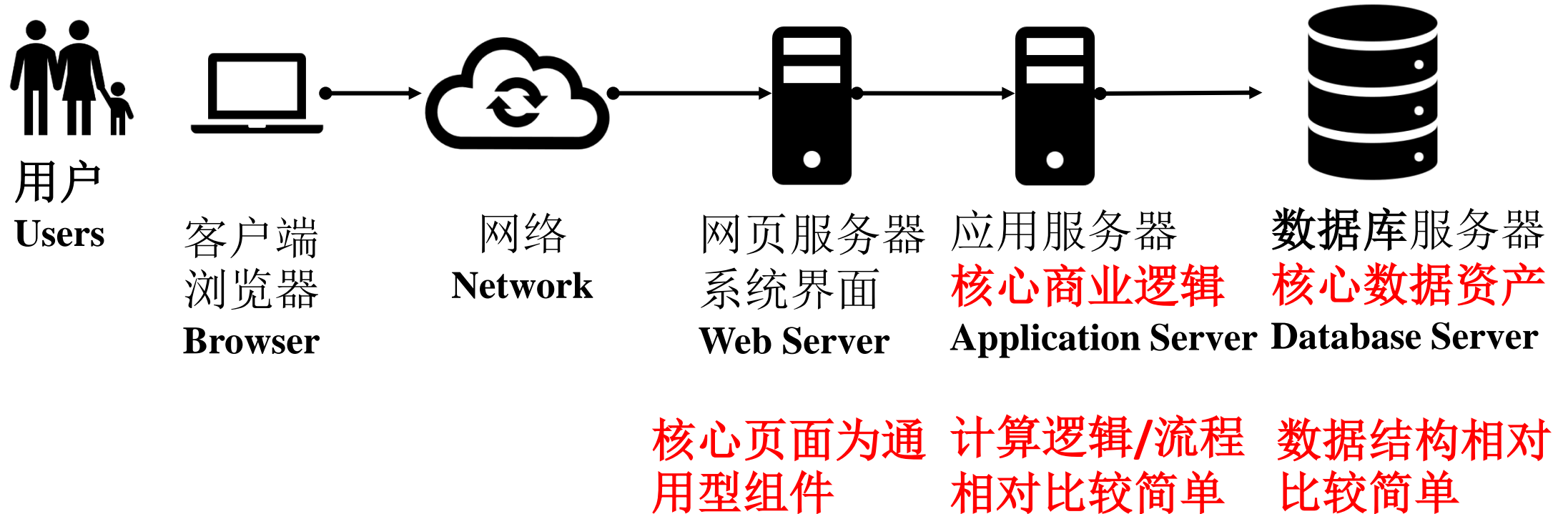




默安科技使用低代码平台

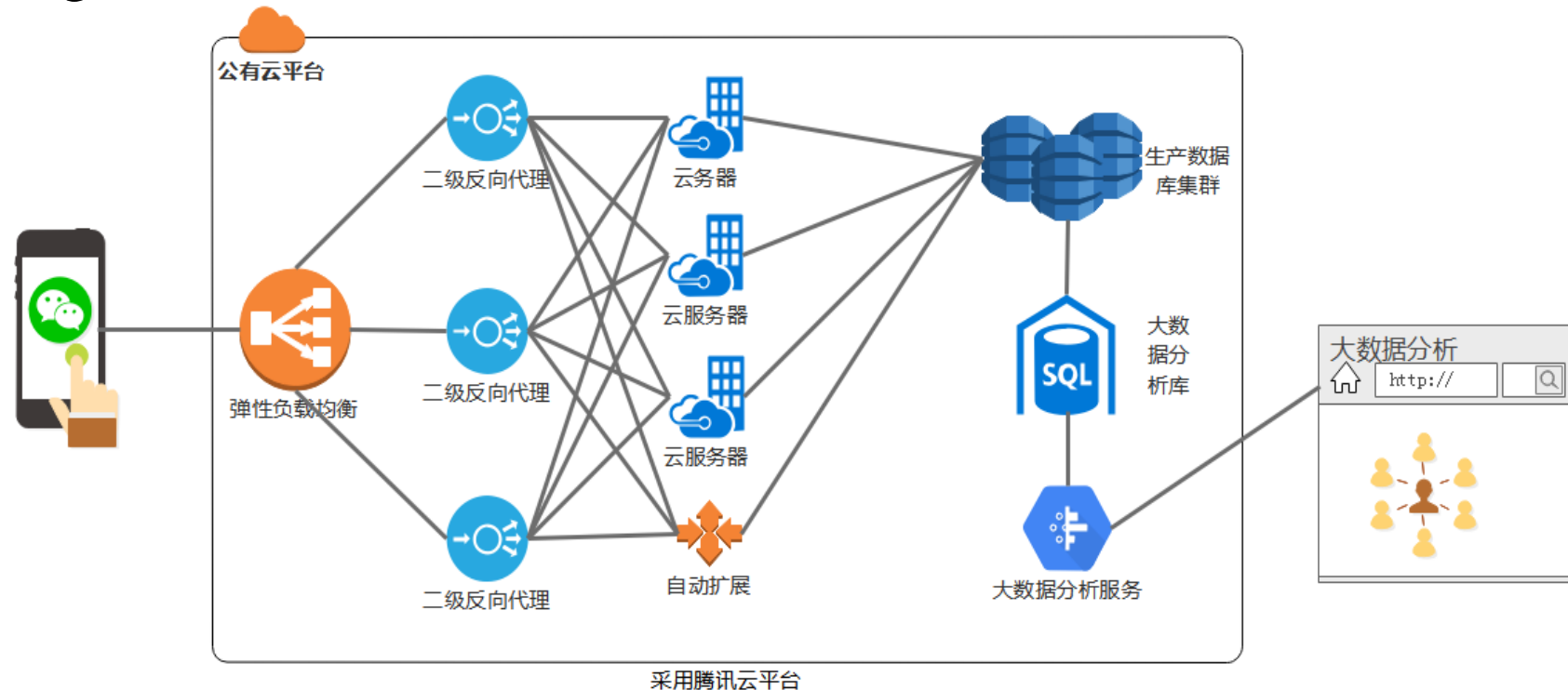
Scenarios applicable to LCDP

- The 80/20 Rule: Capable of achieving 80% of functions and applicable to 80% of scenarios.



Case: High concurrency and high response systems

- How can Baidu return query results within 1 second?
- Load balancing service



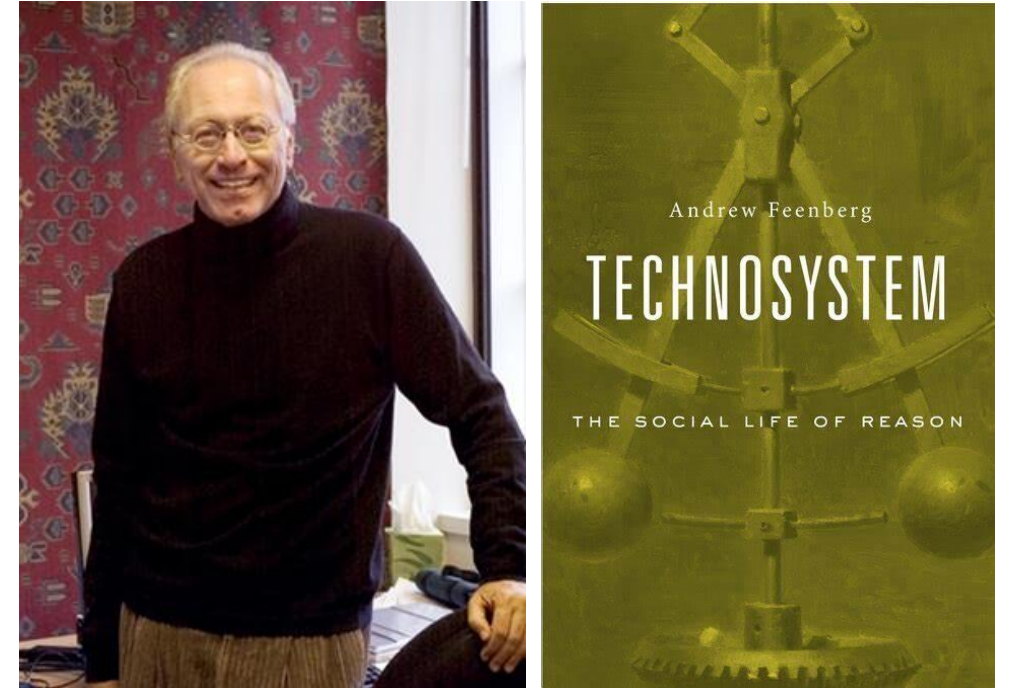
4. 低代码平台的发展趋势

Future Directions of LCDP



1. Democratization of Technology

- Professor Andrew Feenberg, a famous philosopher of Technology, once proposed a very creative concept, "Democratization of Technology."
- Technological democracy is to expand the free boundaries of social individuals, so that everyone can actively and effectively participate in technology design and technology decision-making, including "lay actors" of different identities and different classes, to ensure that their interests can be realized. In short, it is to involve a wide range of people in the design of technology, and ultimately achieve a greater degree of technical collaboration.



Andrew Feenberg (2017). *Technosystem: The Social Life of Reason*: Cambridge, MA: Harvard University Press. 235 pp. ISBN 9780674971783 (Hardcover)

-
- Democratization of Technology means that technology is easily accessible to everyone, no matter what they do or where they live.
 - According to Gartner, four key aspects of the democratization of technology will be the latest technology trends in the postmodern world beyond 2022: application development, design, knowledge, data, and analytics.
 - New technologies and improved user experiences will enable people outside the technology industry to access and use technology products and services. The democratization of technology means easy access to technical or business expertise without extensive or expensive training. With the rise of mass developers, this has been widely recognized.

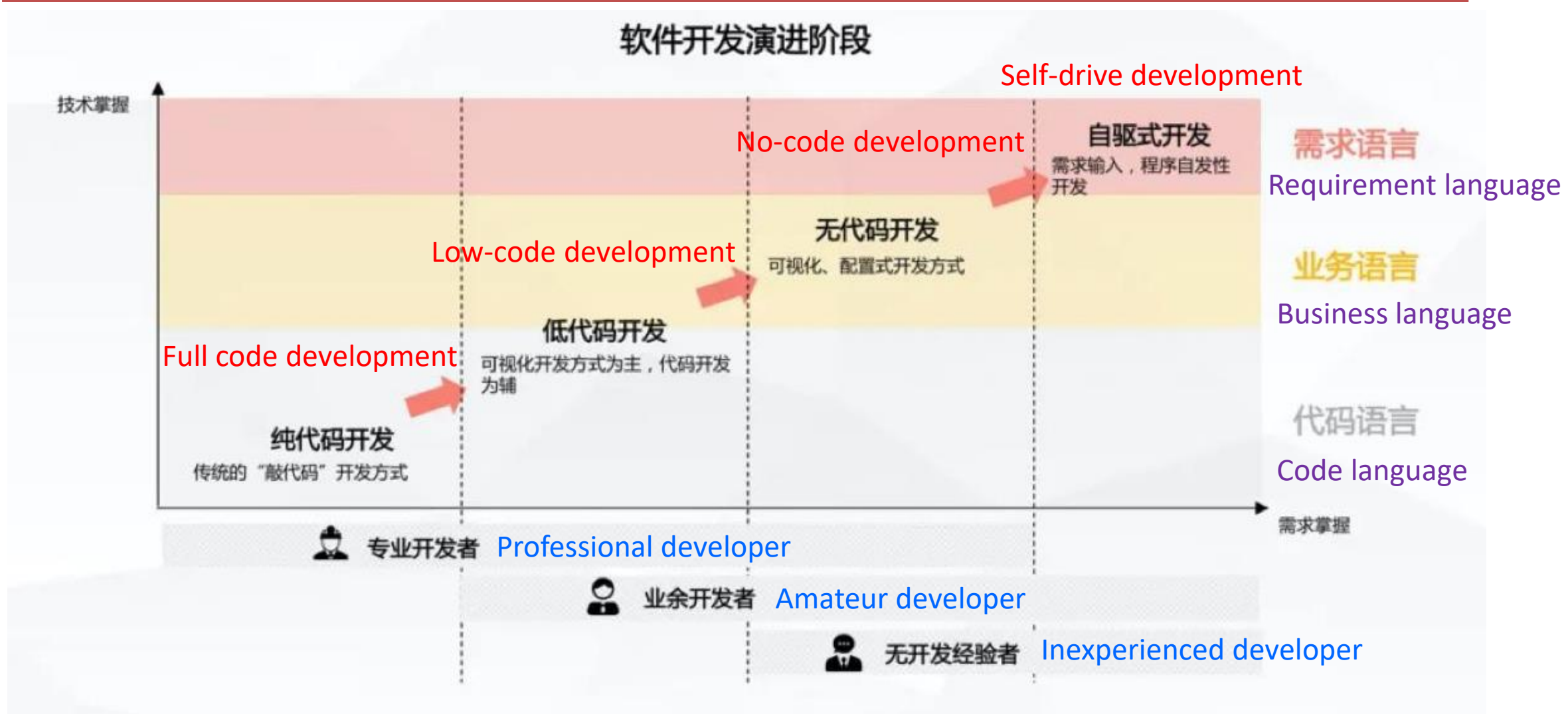
Four aspects of Democratization of Technology

- According to Gartner, from 2020 to 2023, the trend of democratization of technology will see rapid development in the following four areas:
 - Democratization of **data and analytics skills** (tools originally intended for data scientists are rapidly expanding to serve the general developer community).
 - Democratization of **development skills** (using AI tools to enable custom development of applications).
 - Democratization of **design skills** (enabling development through low-code, no-code patterns and empowering developers by building automated development capabilities into applications).
 - Democratization of **knowledge** (non-IT professionals have access to tools and professional-grade systems that allow them to leverage specific skills beyond their own expertise and training).

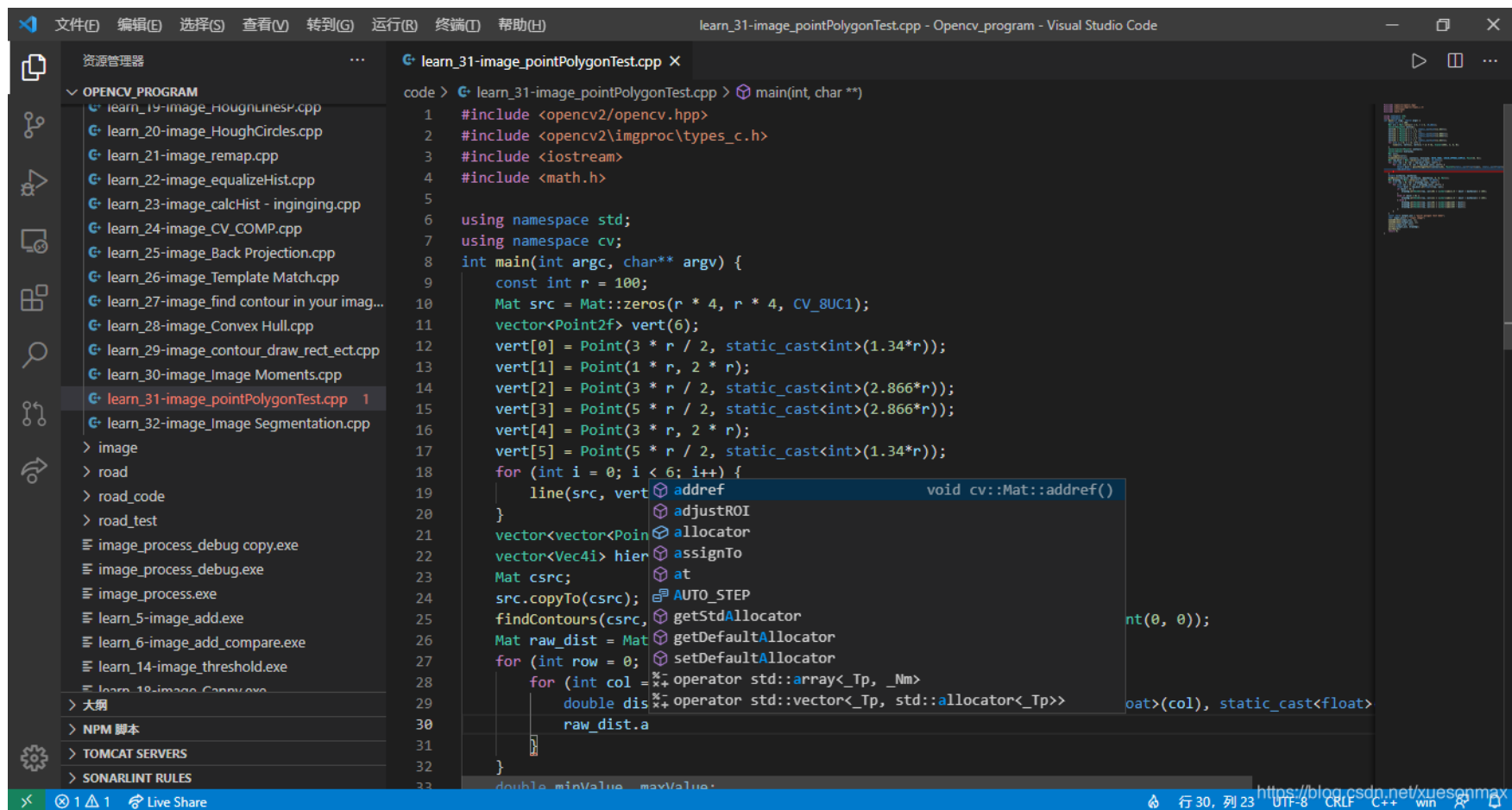
The advantages of democratization of technology

- **Increase employee productivity.** The more access employees have to digital tools and the more skilled they are in using them, the more productive they will be.
- **Promote innovation.** New tools, such as no-code platforms, enable employees to innovate in ways that were not possible before the introduction of these tools. Certain platforms, such as digital adoption platforms, enable employees to automate workflows and even create "automated robots" that interact with employees.
- **Increased agility.** Technology improves efficiency and performance wherever it is applied. The more democratized the technology within an organization, the faster the organization will adopt new tools, move, and transform.
- **Improve flexibility.** Digital disruption is the norm. Organizations that are unable to adopt new technologies and democratize them quickly risk losing market share to companies that can adopt them. On the other hand, organizations that have built a digital adoption plan can integrate new technologies more quickly and remain resilient in the face of digital disruption.
- **Improve organizational performance.** Organizations that can turn technology into an asset - and increase its productivity, efficiency, innovation, and agility - will perform better in the marketplace.
- **Improve customer satisfaction.** One of the main goals of the democratization of technology is to increase customer satisfaction. Technology democracy provides employees with the tools and resources to do their jobs effectively. This will increase productivity and make customers more satisfied.

2. Low-code/No-code is the inevitable trend



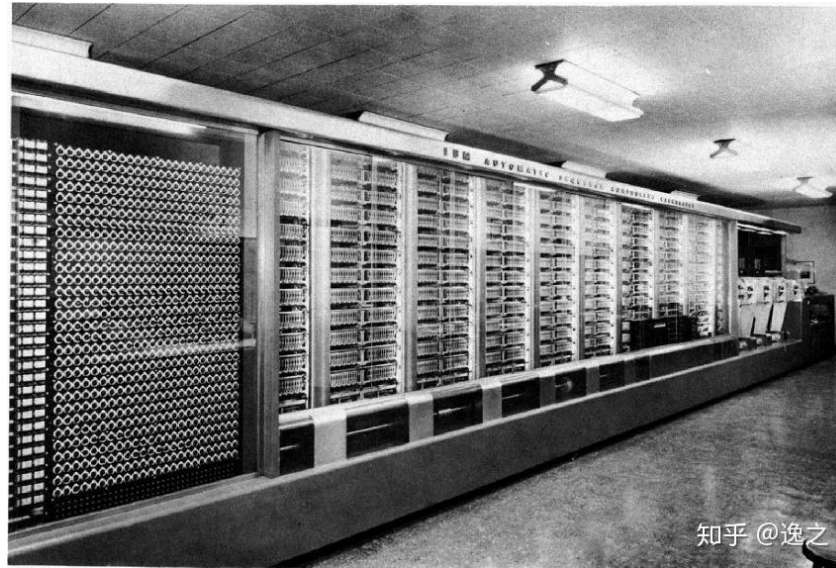
Full-code Programming



```
code > learn_31-image_pointPolygonTest.cpp > main(int, char **)
1  #include <opencv2/opencv.hpp>
2  #include <opencv2/imgproc/types_c.h>
3  #include <iostream>
4  #include <math.h>
5
6  using namespace std;
7  using namespace cv;
8  int main(int argc, char** argv) {
9      const int r = 100;
10     Mat src = Mat::zeros(r * 4, r * 4, CV_8UC1);
11     vector<Point2f> vert(6);
12     vert[0] = Point(3 * r / 2, static_cast<int>(1.34*r));
13     vert[1] = Point(1 * r, 2 * r);
14     vert[2] = Point(3 * r / 2, static_cast<int>(2.866*r));
15     vert[3] = Point(5 * r / 2, static_cast<int>(2.866*r));
16     vert[4] = Point(3 * r, 2 * r);
17     vert[5] = Point(5 * r / 2, static_cast<int>(1.34*r));
18     for (int i = 0; i < 6; i++) {
19         line(src, vert[i], vert[(i+1)%6], Scalar(255));
20     }
21     vector<vector<Point2f>> hier;
22     vector<Vec4i> hier;
23     Mat csrc;
24     src.copyTo(csrc);
25     findContours(csrc, hier, Mat(), CV_RETR_CCOMP, CV_LIST_APPROX_SIMPLE);
26     Mat raw_dist = Mat_<float>(src.rows, src.cols, CV_32F, Scalar_<float>(0));
27     for (int row = 0; row < raw_dist.rows; row++) {
28         for (int col = 0; col < raw_dist.cols; col++) {
29             double dist = 1000000;
30             for (int i = 0; i < hier.size(); i++) {
31                 for (int j = 0; j < hier[i].size(); j++) {
32                     Point2f p(hier[i][j].x, hier[i][j].y);
33                     double d = sqrt((p.x - col) * (p.x - col) + (p.y - row) * (p.y - row));
34                     if (d < dist) dist = d;
35                 }
36             }
37             raw_dist.at<float>(row, col) = dist;
38         }
39     }
40     imshow("Distance Transform", raw_dist);
41     waitKey(0);
42 }
```

哈佛机 1944: 穿孔纸带上的编程 Machine Language-Havard Mark I

- The Mark I is a large computer consisting of about 765,000 electromechanical components with a total length of 800 kilometers of internal wires. The machine is about 15.5 meters long, 2.4 meters high and weighs 5 tons, filling the entire wall of the machine room.
- Programming language designer: Howard Hathaway Aiken



操作	编程代码	穿孔
S10+C24 (结果存入C24)	752 54 7	○●○●○●○●○○○●●○○○●○○○○○
用1号打字机打印S10	752 7432	○●○●○●○●○●○○●●○○○○○○○○
清空C24	54 54 7	○○○●○○○○○○○●○○○○○●○○○○○
关闭2号打字机	8731	●●○○○●○●○○○○○○○○○○○○○○
求C64的正弦值	7 7631	○●○○○○○○○●○○○○○○○○○○○○○○

汇编语言

Assembly Language

- Assembly language, the second generation of computer languages, replaces specific instructions with abbreviated words that are easy to understand and remember, such as: With "ADD" for addition instruction, "SUB" for subtraction instruction, "INC" for increment 1, "DEC" for subtraction 1, "MOV" for variable passing, etc., in this way, people can easily read the completed program or understand the function that the program is performing. bug fixes to existing programs and operational maintenance are made easier.

```
C_S SEGMENT
    ASSUME CS: C_S, DS: C_S
S_T:
    MOV AX, C_S
    MOV DS, AX
    LEA DX, P_S
    MOV AH, 9
    INT 21H
    MOV AH, 4CH
    INT 21H
P_S DB 'Hello World!', 36
C_S ENDS
    END S_T
```


Development of AI large language model + LCDP

- Learn software templates from various industries and become an expert in business system design; Generate requirement documents by chatting and interacting with customers; Based on documentation, generate code and complete software development.
- Just by taking a picture or writing a sentence, without any line of code, you can automatically generate a business application.
- AI large language model + LCDP, new paradigm of software development; Solves 90% of the software needs of smes.

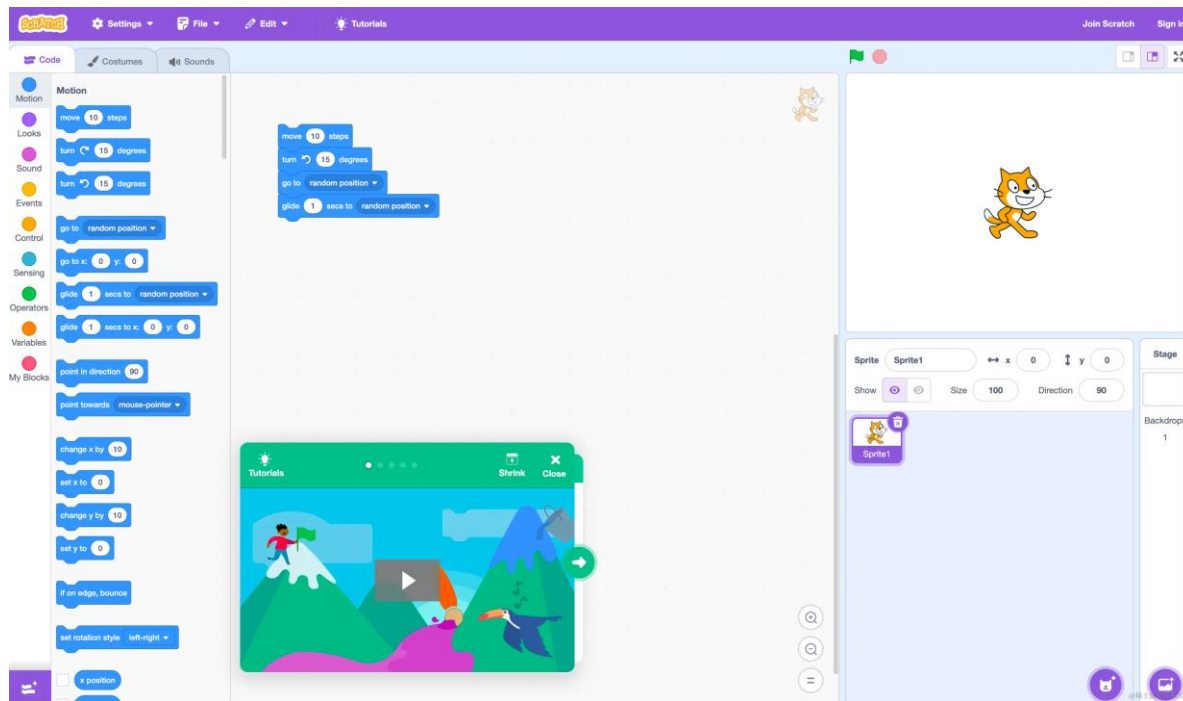
No Code AI

- No-code AI, also called codeless AI, is a category in the AI landscape that aims to democratize AI.
- No-code AI means using a no-code development platform with a visual, code-free, and often drag-and-drop interface to deploy AI and machine learning models.
- A wide range of tools provide no code AI capabilities, including dedicated no code AI tools, as well as some automation tools (e.g. some RPA software providers) that include integrated AI capabilities in a no-code user interface.
- No code AI enables non-technical users to quickly classify, analyze data and easily build accurate models to make predictions.

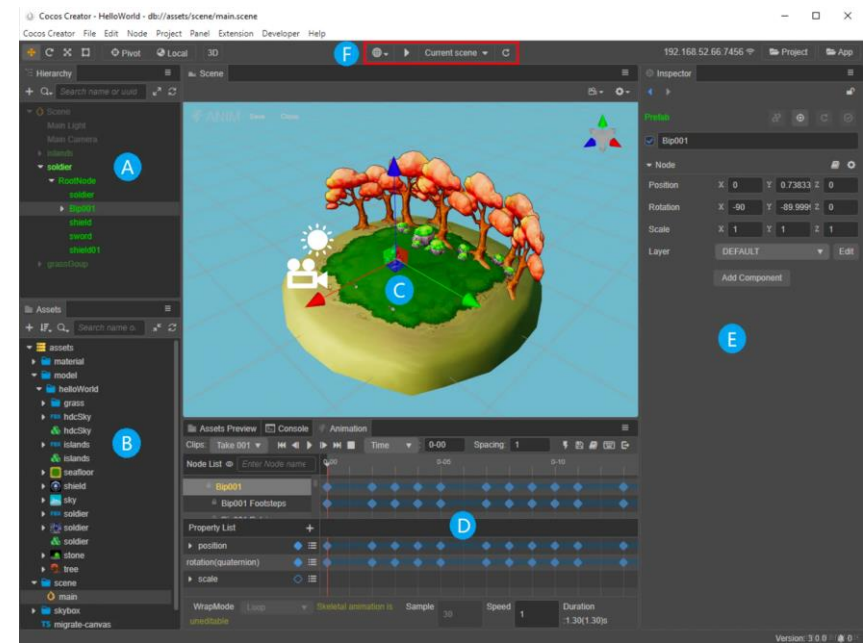
3. 其它一些低代码的例子

Some other examples of LCDP

儿童编程软件 如Scratch
Kids' Programming Software



游戏开发软件 如Cocos Creator
Game Development Tool



调查问卷和数据采集 如金数据 Survey and Data Collection

推动企业数
提供卓越品
从优秀初创团队到业

免费使用

NIO

行业	场景
教育培训 轻松搭建信息化教育平台, 实现从 0 到 1 的突破	营销工具 助力用户在营销获客、客户跟进、客户留存全流程的自动化与信息化
医疗健康 赋能医院业务人员, 解决医院数据管理难题	问卷调查 用表单快速在公众号、朋友圈发布你的问卷调查
婚纱摄影 专业获客工具, 实现在线品牌构建	在线考试 在线考试, 自动评分, 考场环境下领先效率的刷题法宝
灵活自由的搭建业务 IT 系统 灵活自由地搭建业务系统, 快速响应企业需求	活动报名 营销活动利器, 从此告别繁琐的活动策划
	在线预约 客户在线预约, 商家在线接单
	在线收款 在线收款, 安全稳定, 极速到账
	在线投票 简单好用的在线投票
	员工体验管理 把员工变成公司产品和服务的推广大使
	客户管理 零代码搭建客户注册、登录、填写、查询系统

高顿教育

教育/教育培训

致力于打造完整的“终身财经教育”生态体系, 利用金数据实现人效提升

人效提升 流程优化

蔚来汽车

汽车/交通运输

通过金数据, 蔚来汽车实现全流程客户体验极致管理

私有化部署 客户体验管理

中国石油

石油

金数据助力中国石油集团测井有限公司实现安全管理效率提升

安全生产管理

业务中台 如致远互联 Business Middle Platform

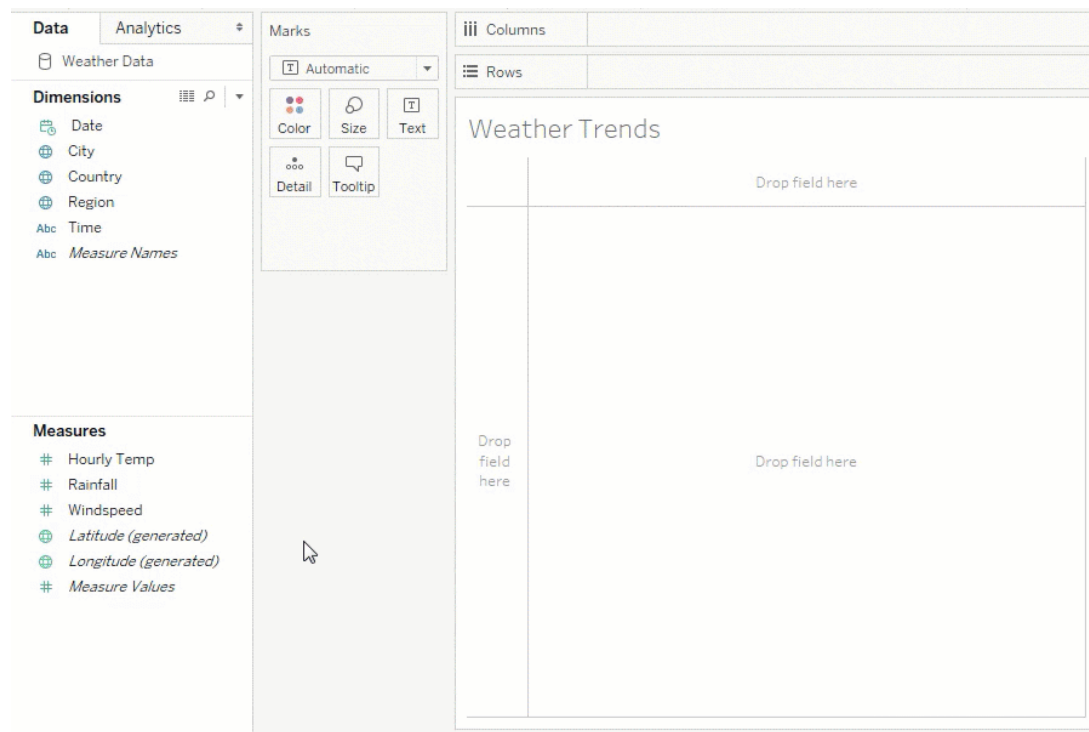
协同工作
业务协同
集成协同
产业链协同
社会化协同

协同工作
实现无纸化办公 提升个人、组织间的协作效率

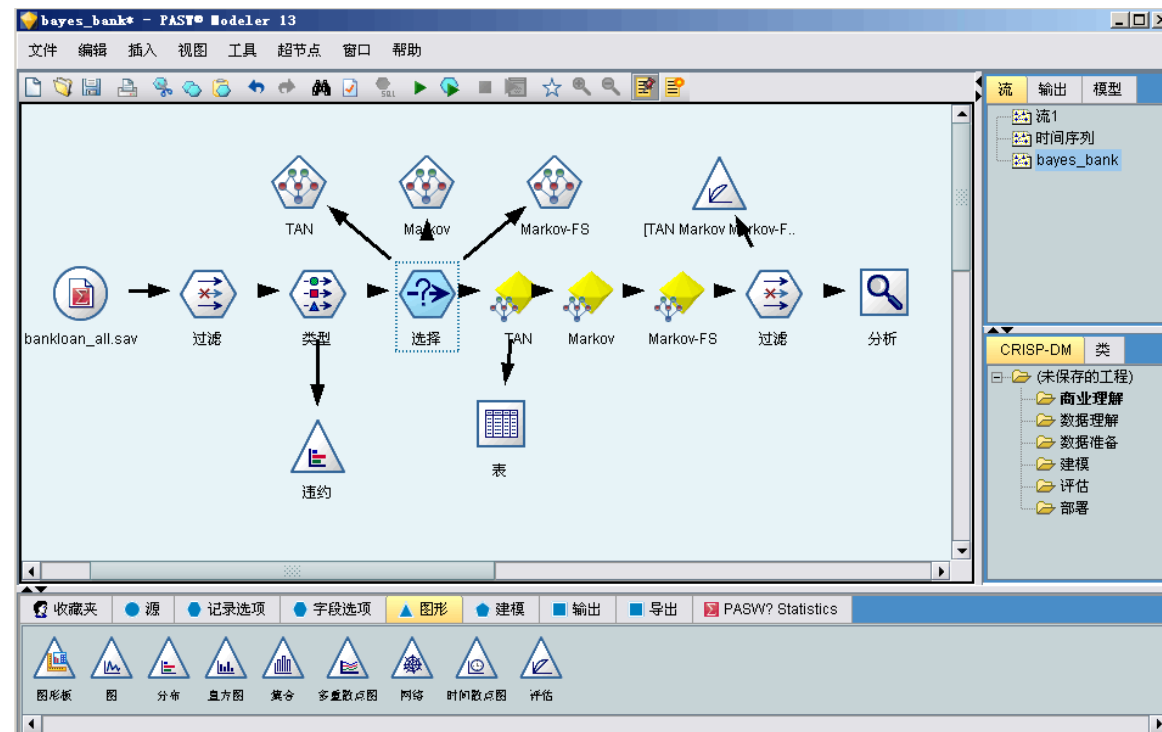
组织管理
协同驾驶舱
知识社区
协作沟通
文化建
会议管理
智能检索
员工自助
计划管理
签到打卡
公文管理

体验产品

数据分析 如Tableau Exploratory Data Analysis



机器学习 如Modeler Machine Learning Modeling



5. 课后作业-1

Homework-1



1. 课后作业 Homework

- 学习使用一个低代码平台 Learn to use a LCDP
- 每个组员分配一个系统中的角色 Each team member is assigned a role in the system
- 模拟业务流转的情况，开发一个系统 Simulate the business flow situation and develop a system
- 画简单的功能结构图、业务流程图 Draw simple functional structure diagram and business flow diagram
- 做PPT汇报+演示系统（10分钟左右） Make PPT presentation + presentation system (about 10 minutes)
- 各小组互相打分 Rate each other

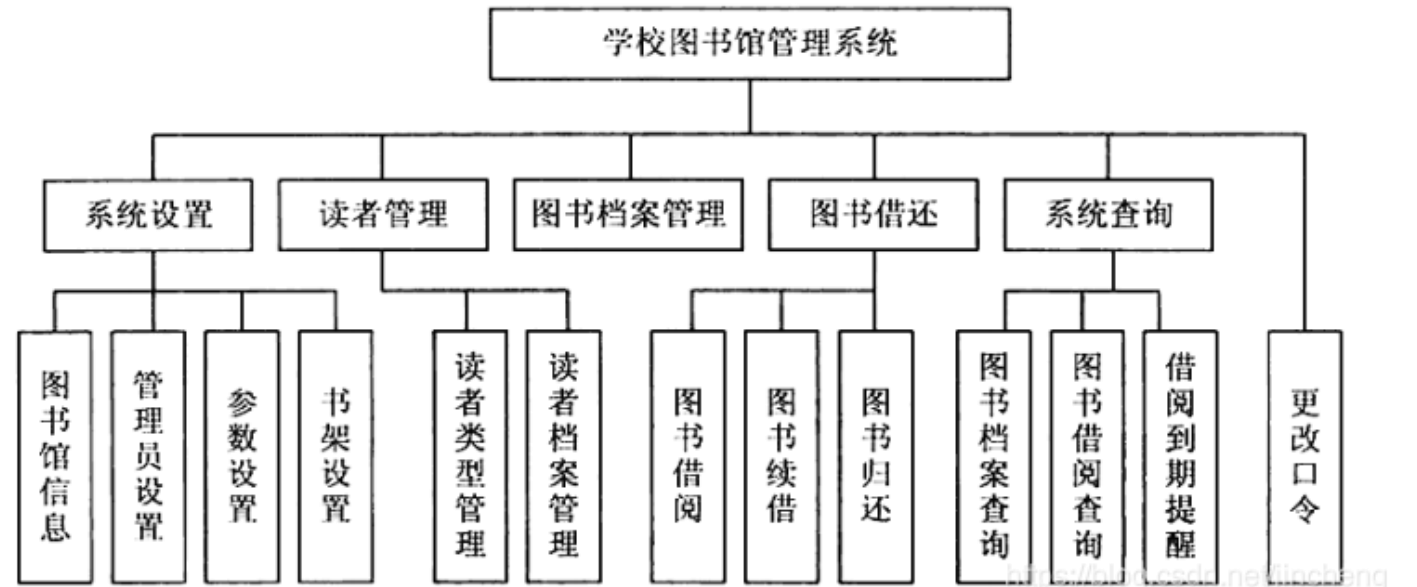
2. 低代码开发平台教程Tutorial of LCDP

- 低代码-宜搭-小课堂（30分钟）small classroom (30 minutes)
<https://www.bilibili.com/video/BV1rp4y1x7cz/>

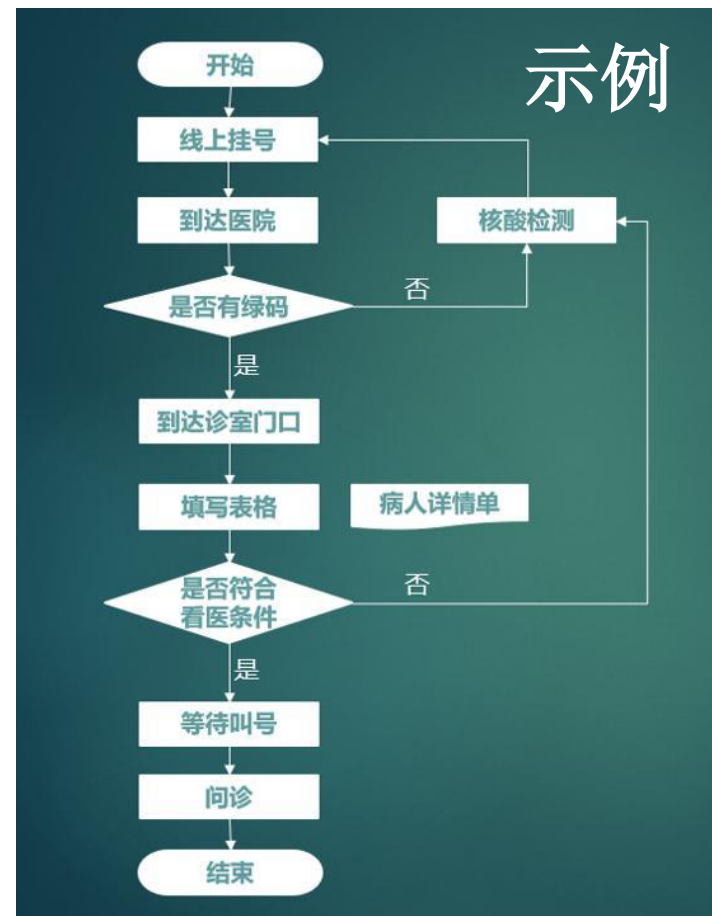


3. 功能结构图functional structure diagram

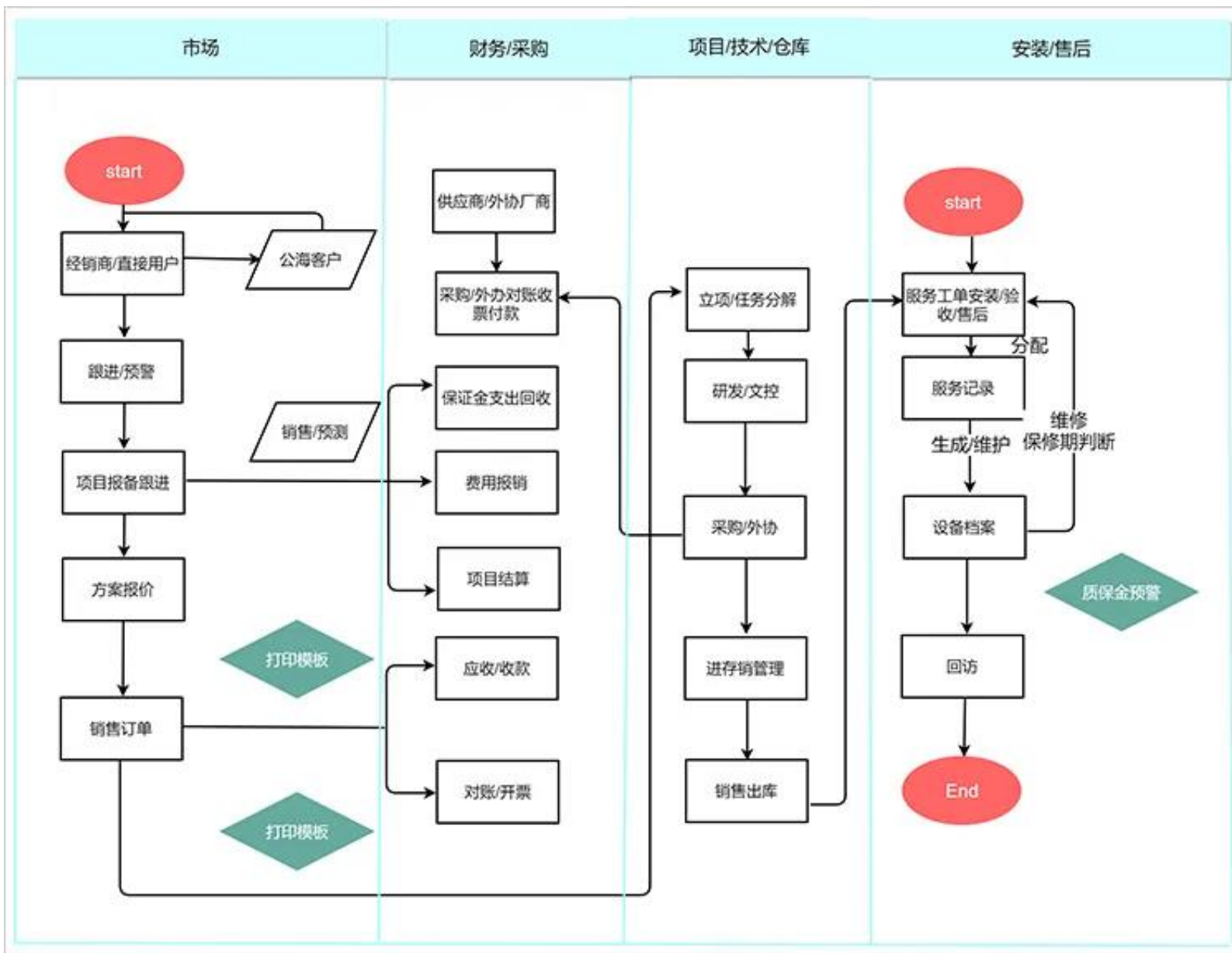
- The function structure diagram is a diagram that decomposes the function of the system and expresses it according to the function dependency relationship. Each subsystem of the management information system can be regarded as the function of the lower level of the system target, and each function can continue to be decomposed into the third layer, the fourth layer... even more features.
- The functional structure is designed to be layered and decoupled.
- Kind of like a mind map.



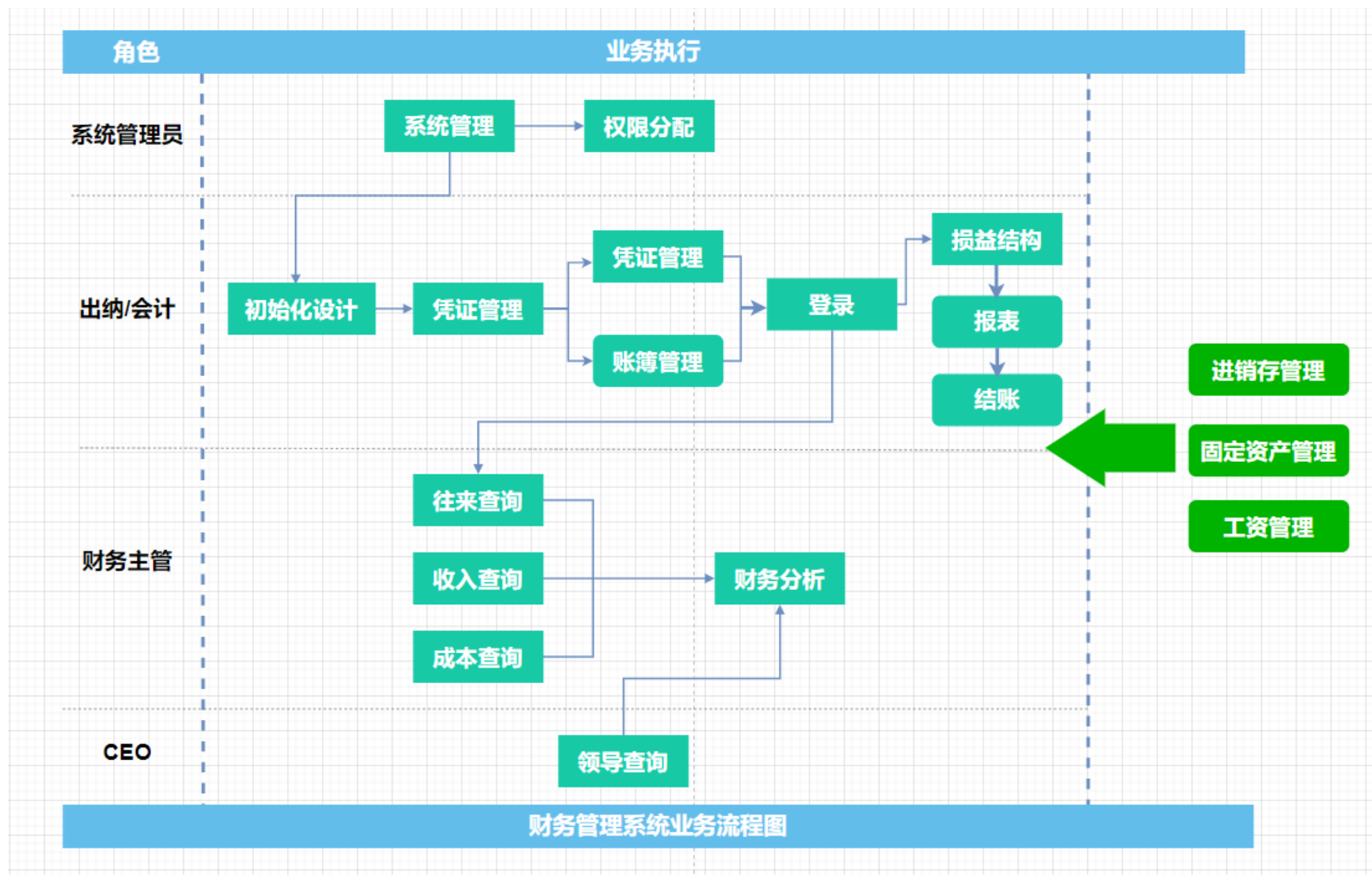
3. 业务流程图business flow diagram



Multi-role business flow diagram

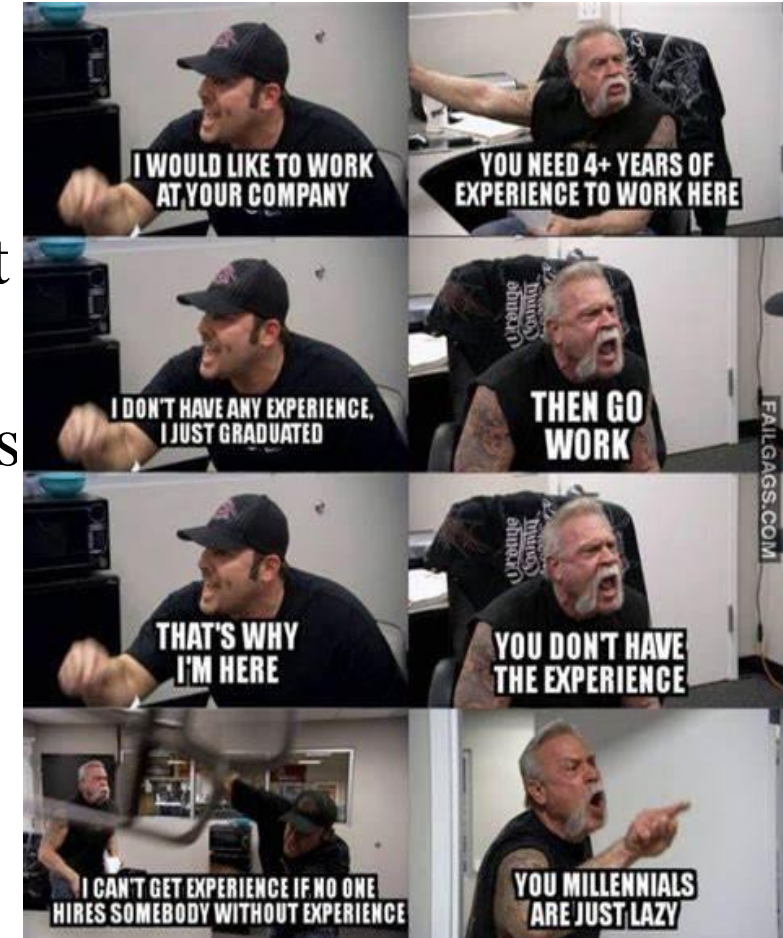


Multi-role business flow diagram



Some explanations of business flow diagram

- There is no fixed format for business flow diagram, but they are required to be illustrative.
- Don't be intimidated by the flow chart, the flow chart is the key!
- Complex, incomprehensible business flow diagram is the problem itself.
- If the process is handled by people, it is likely to cause inefficiency and wrangling.
- What about businesses that are outside the business flow diagram (so-called irregularities)?



谢谢！

Thank you for your attention.

liuyuewen@xjtu.edu.cn

