

软件及其特点



微信扫码签到

内容

1. 软件

✓软件的概念、特点、组成及生命周期

2. 开源软件

✓开源软件实践，托管平台和社区，开源许可证

3. 软件质量

✓软件质量要素及模型

4. 软件特征的变化

✓地位、运行环境、形态、复杂性



建设狗窝和建设大厦的区别

□建设狗窝的特点

- ✓简单
- ✓质量要求不高
- ✓一个人就可完成



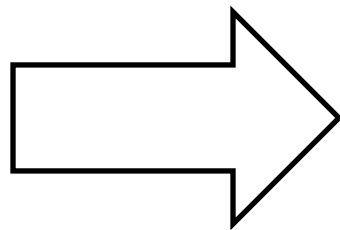
□建设大厦的特点

- ✓复杂
- ✓质量高要求很高
- ✓需要团队合作才能完成



1.1 从程序到软件-工程开发的启发

应用
(如火车票购买、酒店预订、网上购物)



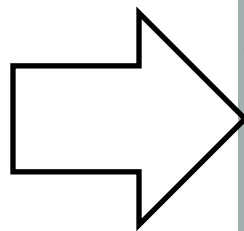
编写程序



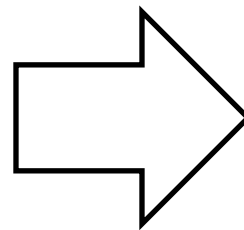
通过一个步骤就将程序写出来不现实



了解需求



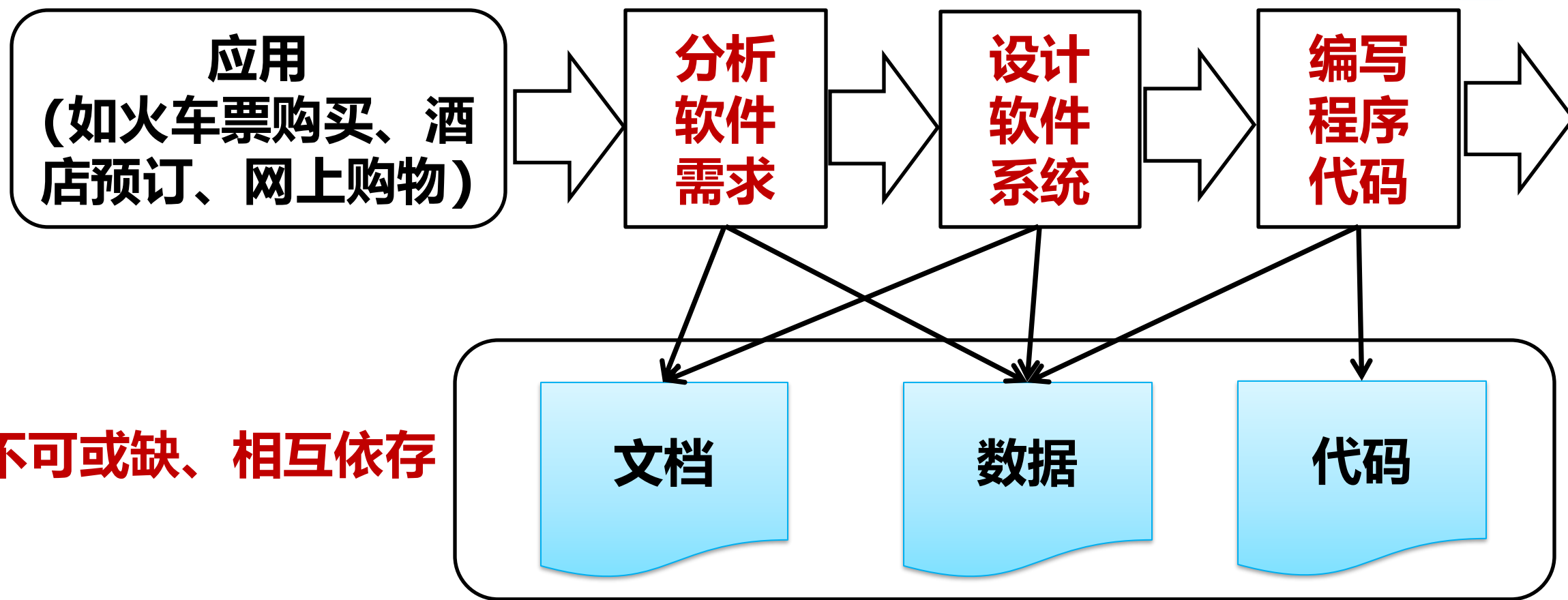
设计图纸



进场施工



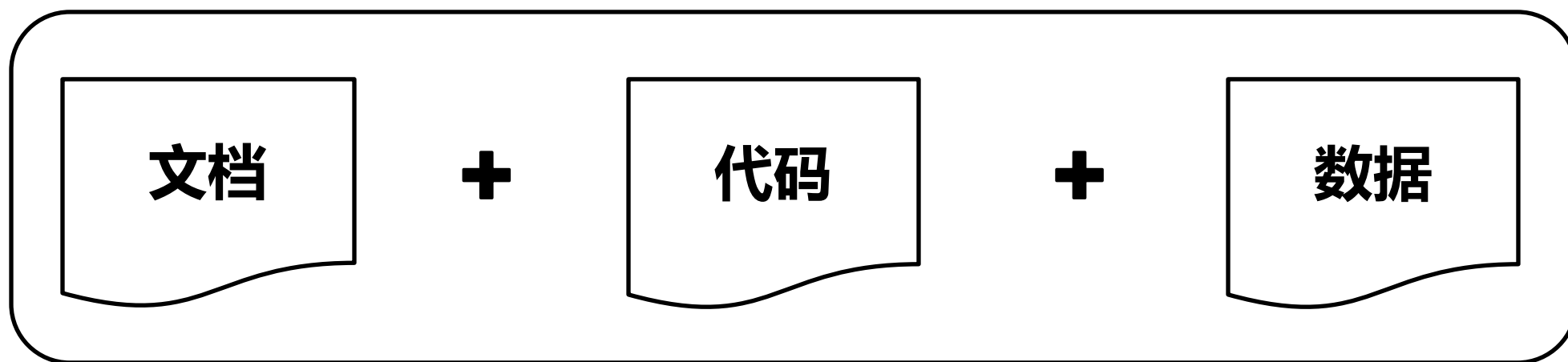
编写出程序需要经历诸多的步骤



要编写出高质量的程序需要循序渐进地开展工作

1.2 何为软件?

□ 软件是指在计算机系统的支持下，能够完成特定功能与性能的程序、数据和相关文档



软件制品
(Software
Artifact)

从开发的角度看软件

何为文档?

□记录软件开发活动和阶段性成果、软件配置及变更的**阐述性资料**

- ✓定义和理解软件
- ✓记录软件开发成果
- ✓辅助不同人员间的交流

- ✓ **软件需求文档**
- ✓ **软件设计文档**
- ✓ **软件测试文档**
- ✓ **软件用户手册**
- ✓

◀ “空巢老人看护系统” 软件系统的需求构思及描述 ◀

阐述性资料

1. 背景介绍◀

随着人口老龄化问题越来越严重，空巢老人的无人看护将成为一个亟待解决的社会难题。家属外出工作时，经常会惦记独自在家的老人，希望掌握老人的状况（比如：身体状况如何？在做什么？）。但是由于空间距离上的限制，家属不能方便快捷的获取这些信息；很多老年人不会或者不方便使用手机，造成家属与老人之间语音通话上的困难；老年人记忆力普遍衰退，容易忘记服药等一些重要的事情，家属外出时无人监督老人完成这些事情。对于一些患有疾病的老人来说，忘记服药很可能会造成危险；当老人独自在家时，他们可能会遇到摔倒或者中风等突发情况，而外出工作的家属和医疗救护人员不能及时掌握这些危险信息，导致救援不及时。◀

本软件专为解决空巢老人的看护问题而设计，它可以帮助用户（老人的家属）监控老人在家的状况，帮助老人与家人语音聊天，提醒并监督老人按时服药，并在发生危险时向用户警报或者联系急救部门。◀

为什么需要文档?

□原因

- ✓ 软件开发牵涉编程之外的**多方面工作**
- ✓ 任何事都记在脑子里, **记不住**
- ✓ 任何事都放在脑子里, **理不清**
- ✓ 任何事都存在脑子里, **讲不明**

□编写文档目的

- ✓ **阐述清楚**: 内容、逻辑性、条理性
- ✓ **发现问题**: 完整、一致、矛盾
- ✓ **开展交流**: 便于各种人员的交流
- ✓ **促进管理**: 管理软件开发成果



空巢老人智能看护系统设计文档







1、引言.....	4
1.1 编写目的.....	4
1.2 读者对象.....	4
1.3 软件项目概述.....	4
1.4 文档概述.....	5
1.5 定义.....	5
1.6 参考资料.....	5
2、软件设计约束.....	5
2.1 软件设计目标和原则.....	5
2.2 软件设计的约束和限制.....	6
3、软件设计.....	6
3.1 软件体系结构设计.....	6
3.2 用户界面设计.....	9
3.3 用例设计.....	13
3.4 类设计.....	24
3.5 数据设计.....	32
3.6 部署设计.....	33

何为数据？

□数据是程序的加工处理对象和结果

- ✓需要处理哪些数据
- ✓如何来获得这些数据
- ✓如何来表示这些数据
- ✓如何来存储和检索数据
- ✓如何来传输数据



		
Apple iPhone 11 (A2223) 128GB 白色 移动联通电信4G手机 双卡双 ¥5899.00	华为 HUAWEI Mate 30 5G 麒麟 990 4000万超感光徕卡影像双超级 ¥4499.00	Apple iPhone 11 Pro Max (A2220) 256GB 暗夜绿色 移动联通电信4G ¥10899.00
		
华为 HUAWEI Mate 30 Pro 5G 麒麟 990 OLED环幕屏双4000万徕卡 ¥5899.00	【向往的生活同款】小米10 双模 5G 骁龙865 1亿像素8K电影相机 ¥4349.00	Redmi K30 5G双模 120Hz流速屏 骁龙765G 前置挖孔双摄 索尼6400 ¥1599.00

网上商城中的手机及其数据

数据示例

□数据形式

✓用户、订单、交易、日志
数据

□数据处理

✓表示、获取、存储、检索、
分析

华为 (HUAWEI) 华为Mate20 手机 麒麟980AI智能芯片全面屏超微距影像超大广角 亮黑色 全网通 6GB+64GB

6.53英寸, 4000毫安电池

京 东 价 **¥ 3999.00** 降价通知 累计评价 100+

优 惠 券 **满2000减19** **满1599减15** **满999减9** 更多>>

增值业务 高价回收, 极速到账 3元1G

配 送 至 有货 支持 送运费险 | 一年质保 | 全国联保 | 7天价保

由 亿品汇旗舰店 负责发货, 并提供售后服务.

选择颜色  极光色  亮黑色  宝石蓝  翡冷翠  樱粉金

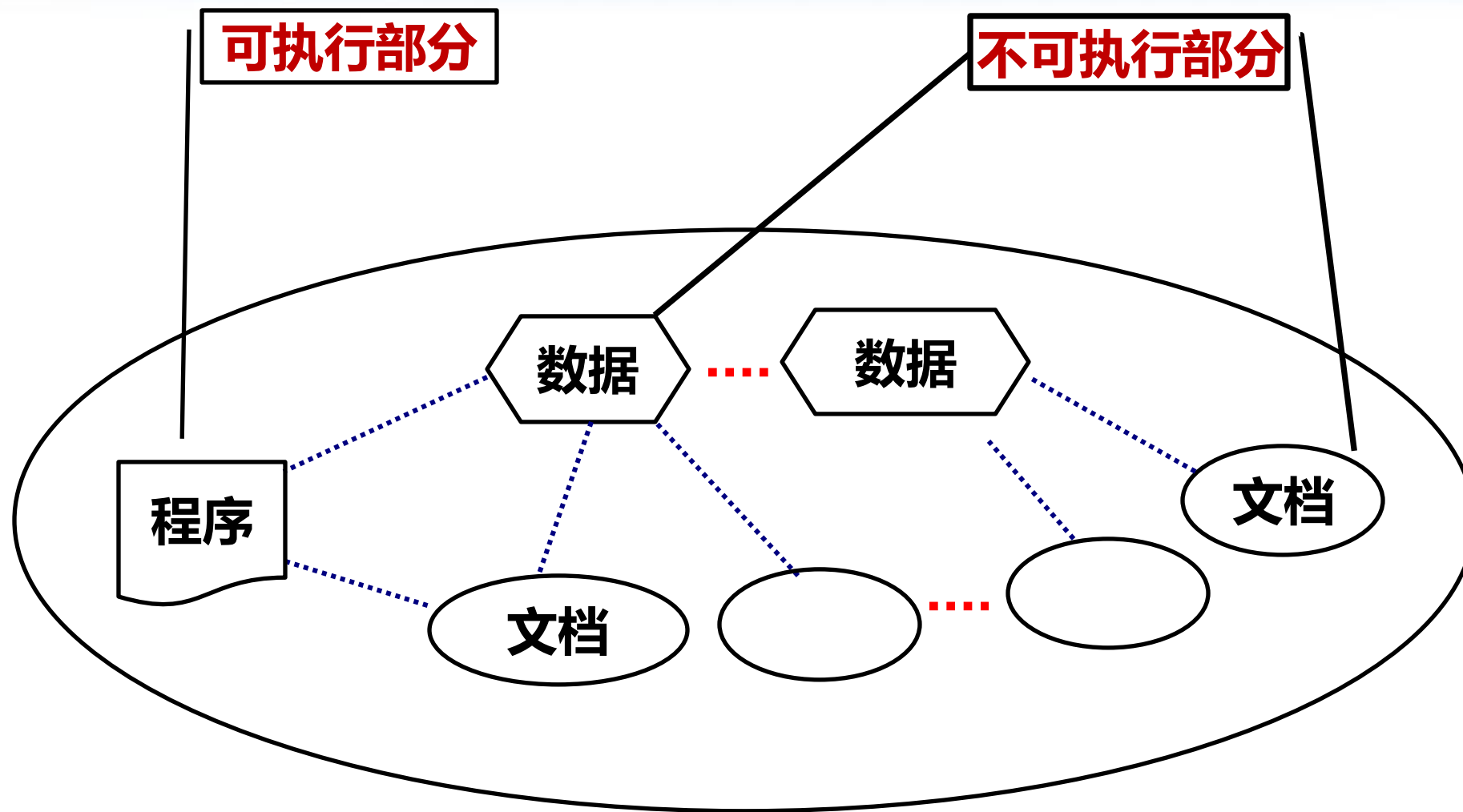
选择版本 全网通6GB+128GB

增值保障  2年全保修 ¥199  1年碎屏保修 ¥99  2年京享无忧 ¥249 

白条分期 

  **加入购物车**

从开发者的角度看软件的构成



1.3 软件概念给我们的启示

□ 软件 \neq 程序

- ✓ 软件涉及更多制品
- ✓ 这些制品间有关联、相互依赖

□ 开发软件 \neq 编写程序

- ✓ 软件开发需要做更多事情
- ✓ 对软件提出更高的要求

软件开发的再认识

□开发具有一定规模和复杂性的软件系统与编写一个简单的程序不一样

✓建设简单平房 vs 建设高楼大厦

✓挖一条小沟 vs 建设方一条南水北调渠道

软件的规模和复杂性意味着要采用行之有效的方法—工程方法

软件开发的“辛苦”加班

- 某天下午5点的时候，主管发消息给我。主管：晚上加个班，版本验证了再走。我：好的
- 无聊等到了晚上8点30。我：x哥，版本什么时候可以用？
- 主管：再等等，刚才有个XX紧急合入，重新跑构建了。我：好的。
- 继续无聊等到了晚上10点30。我：x哥，版本出来了没有？
- 主管：再等等，编译出错，代码回退了，正在跑构建。我：好的。
- 继续无聊等到晚上12点。我：x哥，版本可以用了吧？
- 主管：再等等，今晚上一定要版本验证。我：好的。
- 继续无聊等到凌晨1点20。主管：邦彦，你早点回去休息吧，版本出不来了。
- 我：。。。。。（不好了）

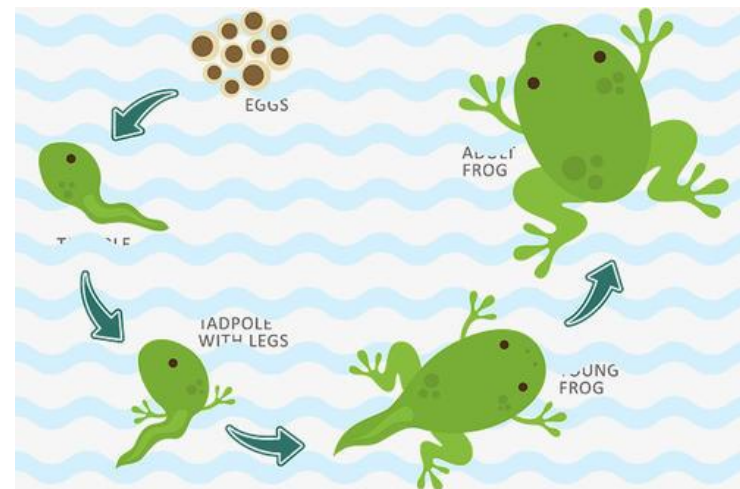
□ 作者：邦彦

□ <https://www.zhihu.com/question/609175609/answer/3094619715>

1.4 软件生命周期

□万物均有生命周期(Lifecycle)

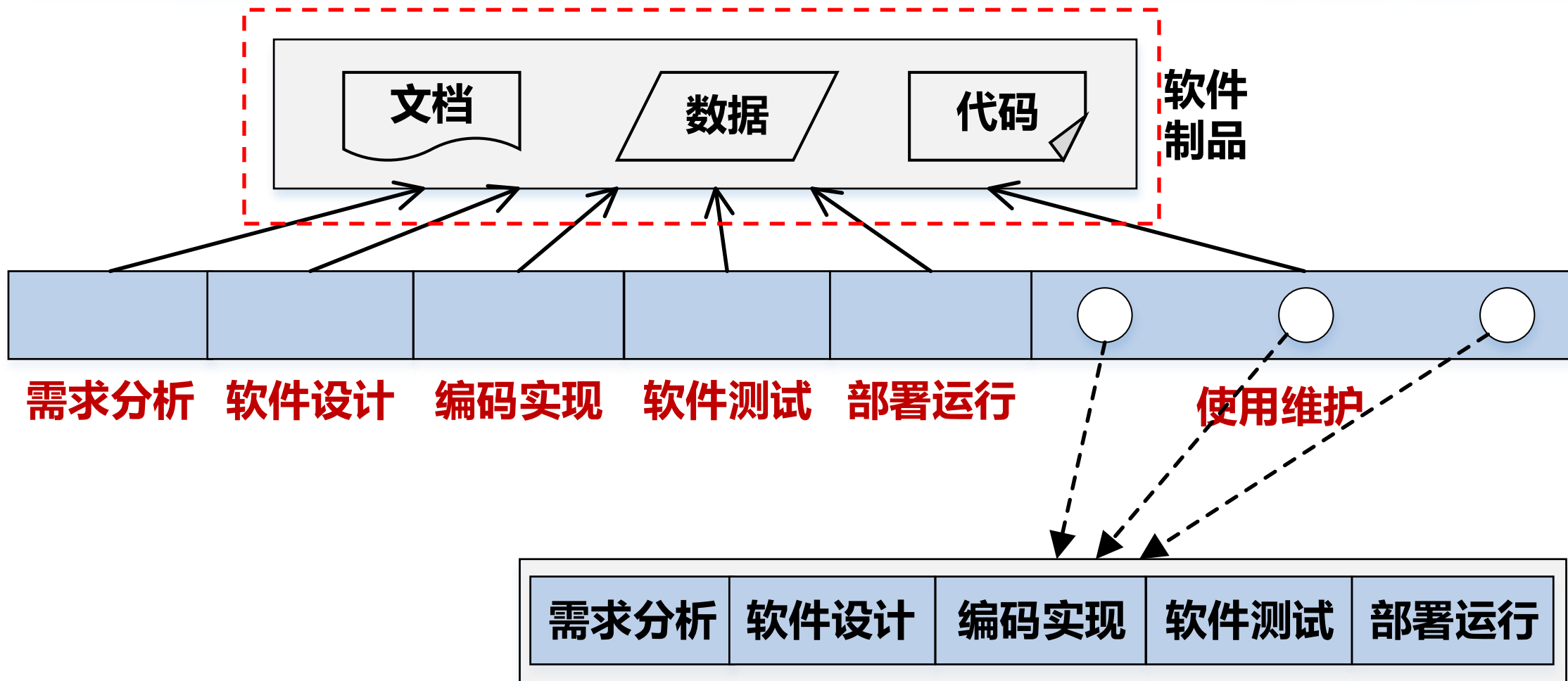
- ✓人：婴儿、幼儿、儿童、少年、...
- ✓树：发芽、育苗、结果、枯萎、死亡、....
- ✓青蛙：受精卵、蝌蚪、去掉尾巴的青蛙 ...



□软件生命周期

- ✓软件从提出开发开始到最终灭亡所经历的时期
- ✓不同阶段具有不同的特征

软件生命周期



从提出开发开始到开发出系统、运行维护以及最终退役的全过程

软件的生命周期

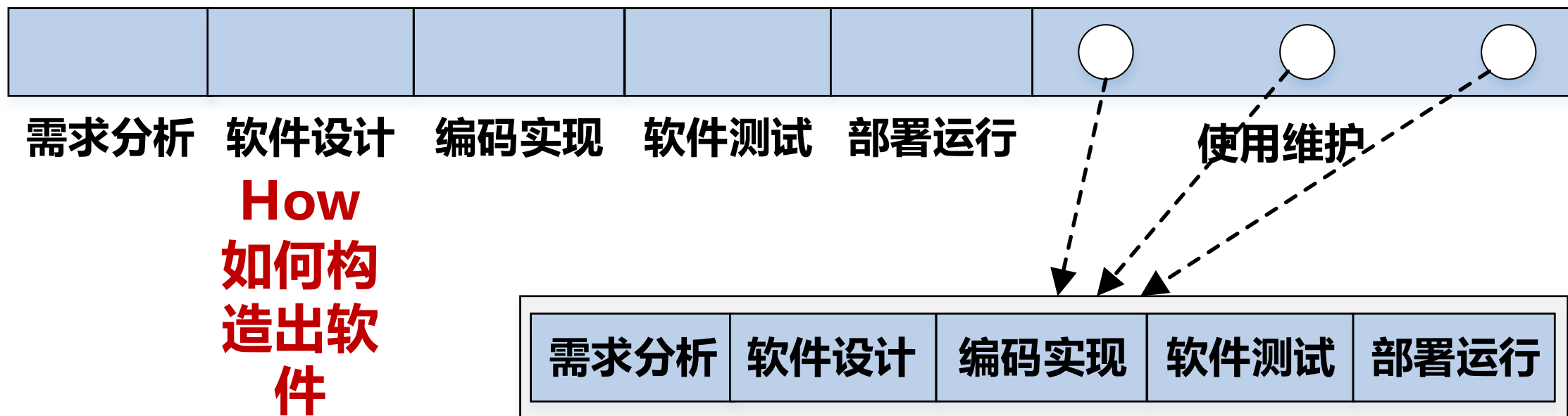
What
软件需求是什么

How
编写程序代码

软件是否缺陷

部署软件运行

每个阶段会产生不同的软件制品



1.5 软件特点

□ 逻辑性

- ✓ **逻辑产品**，思维活动（而非物理活动）的结果、不会磨损和老化

□ 设计开发

- ✓ 是**设计开发**而成的，不是生产制造而成的

□ 易变性

- ✓ 需求**经常变**、难以把控，影响软件的制品即开发过程

□ 复杂性

- ✓ **规模大**：代码行、模块、接入人员、进程、数据等数量非常大
- ✓ **运行复杂**：状态很难追踪和复现

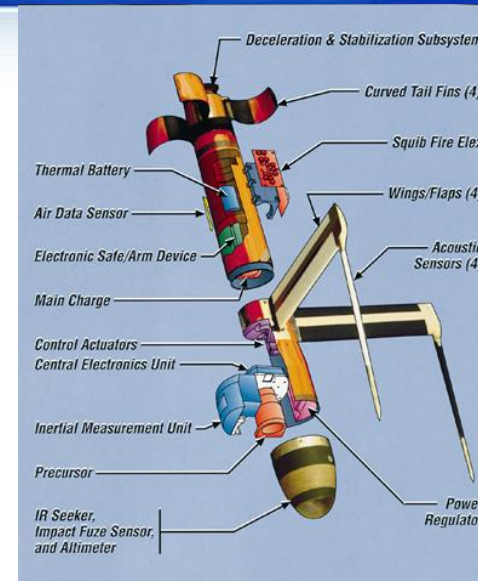
□ 缺陷的隐蔽性

- ✓ 缺陷**隐藏在逻辑代码**中，不像硬件系统那样直观显现，很难被人们所发现和排除

军用软件的特殊性

- 运行环境更复杂和多样
- 与物理系统(如飞机)紧密联系在一起
- 对实时性和质量(如可靠性)提出更高要求
- 对自身防护能力(如安全性)提出更高要求
- 需具备更强的灵活性、适应性和健壮性

军用软件：导弹中的飞行控制软件、指挥信息系统、后勤保障软件等



1.6 软件分类

□ 应用软件

- ✓ 面向**特定应用领域**的专用软件。它们针对相关行业和领域的特定问题，为其提供基于计算的新颖解决方案。
- ✓ 示例：淘宝、12306、携程、微信、QQ等软件

□ 系统软件

- ✓ 对计算机资源进行管理，**为应用软件的运行提供基础设施和服务**的一类软件。从计算服务的视角，系统软件介于计算机硬件和应用软件之间。
- ✓ 示例：操作系统、数据库管理系统、编译软件、软件中间件等

□ 支撑软件

- ✓ **辅助软件开发和运维**，帮助开发人员完成软件开发和维护工作的一类软件
- ✓ 示例：SonarQube、Visual Studio、Eclipse等

软件分类

类别	服务对象	软件的功能	发挥的作用
应用软件	行业和领域应用的用户	为特定行业和领域问题解决提供基于软件解决方案，创新应用领域的问题解决模式	提供更为便捷、快速、高效的服务
系统软件	各类应用软件	为应用软件运行和维护提供基础设施和服务，如加载、通讯、互操作、管理等	作为应用软件的运行环境
支撑软件	软件开发者和维护者	为软件系统的开发和维护提供自动和半自动的支持	提高软件开发效率和质量

内容

1. 软件

✓软件的概念、特点、组成及生命周期

2. 开源软件

✓开源软件实践，托管平台和社区，开源许可证

3. 软件质量

✓软件质量要素及模型

4. 当前软件特征的变化

✓地位、运行环境、形态、复杂性



2.1 闭源软件

□何为闭源软件

- ✓ 软件代码不对用户开放的一类软件，购买软件时只提供可运行软件或服务，没有提供源代码
- ✓ 以许可证（License）的方式授权用户使用软件

□闭源软件带来的问题

- ✓ 无法掌握软件内部实现情况（如是否存在安全漏洞和恶意代码），也无法对软件进行修改和完善，极大影响了开发者的创新自由

□示例

- ✓ 微软的Windows、Office软件，
- ✓ 典型企业：微软、IBM、Oracle等



2.2 开源软件

□何为开源软件

- ✓一种源代码可以**自由获取和传播**的计算机软件，其拥有者通过开源许可证赋予被许可人对软件**进行使用、修改和传播**

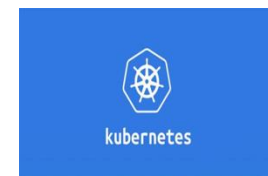
□开源软件的好处

- ✓源代码可自由传播
- ✓激发创作者的热情
- ✓免费使用降低成本



示例：开源软件

- ❑ Linux, Ubuntu
- ❑ Eclipse
- ❑ Apache, Jboss, Tomcat
- ❑ Kubernetes(k8s)
- ❑ MySQL
- ❑ Firefox
- ❑ Google:TensorFlow



Github上Linux

Repositories	138K
Code	
Commits	1B
Issues	2M
Marketplace	1
Topics	563
Wikis	192K
Users	15K

- ▶ Apollo Auto, 百度推出的自动驾驶开放平台和灵活架构
- ▶ ONAP, AT&T 创建的开放网络自动化平台
- ▶ Monica, 管理个人交际关系的开源软件

你还能举出哪些开源软件？



开源软件正逐步替代闭源软件

□服务器操作系统领域

- ✓Linux、FreeBSD为内核的操作系统逐步替代Unix

□桌面操作系统领域

- ✓Linux为核心的开源操作系统正替代Windows操作系统

□数据库领域

- ✓MySQL开源软件受到越来越多用户的青睐，逐步代替Oracle

□浏览器领域

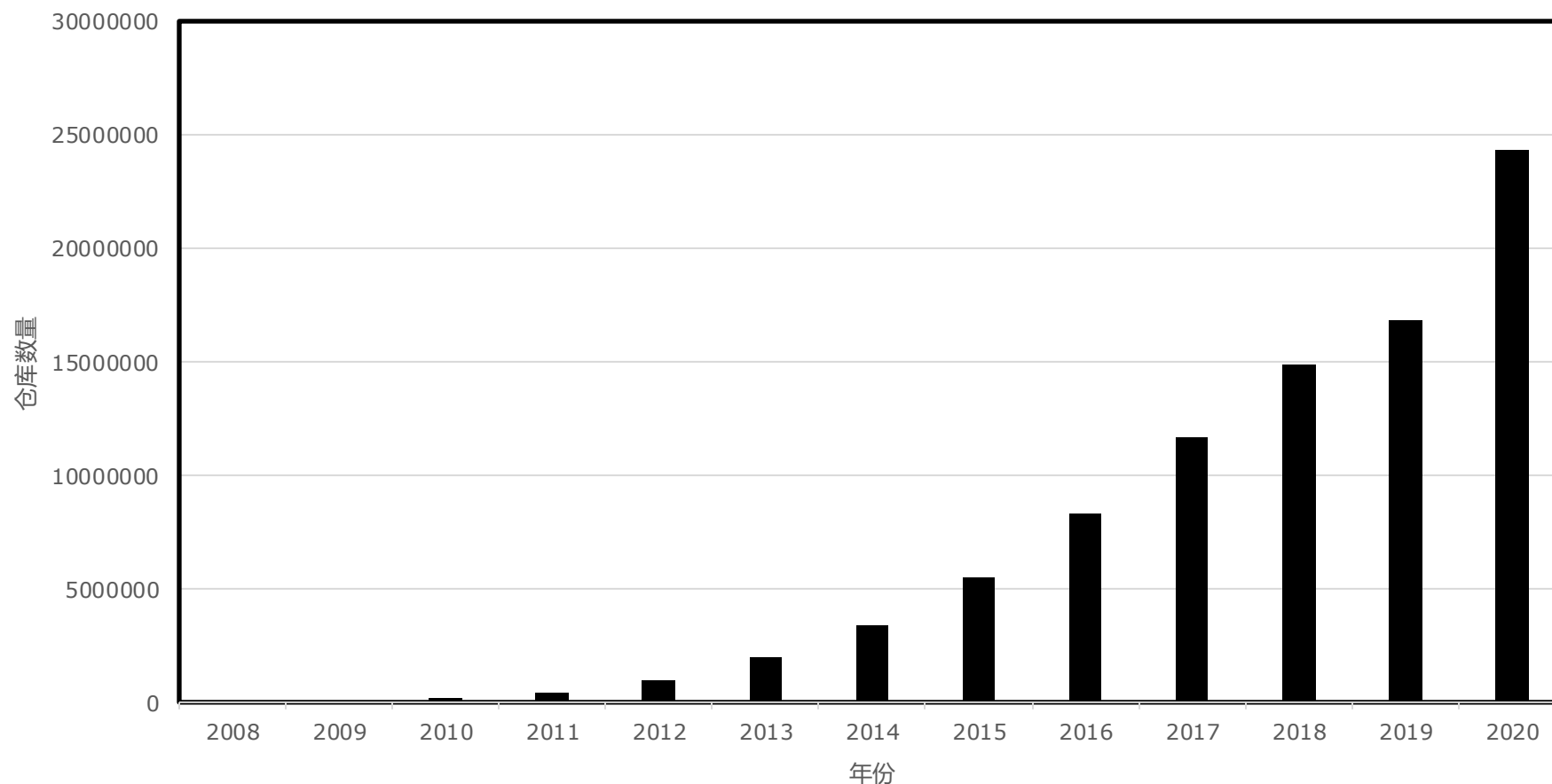
- ✓Chrome和 Firefox开源软件拥有排名第二和第三的市场占有率

□开发工具领域

- ✓Eclipse的影响力日增，成为Java 开发人员的首选集成开发环境

快速增长的开源软件数量

□ Github上的开源软件仓库(www.github.com)



**2021年
Github上有
2亿多的开源
软件仓库**

开源软件的推动者

□政府组织

- ✓ NASA开源200多软件项目，美军鼓励采用开源软件
- ✓ 我国政府将开源列入十四五国民经济和社会发展规划纲要

□开源组织

- ✓ Apache基金会、Linux 基金会、Eclipse基金会、Open Source Initiative

□企业

- ✓ Google、Microsoft、IBM、Oracle等国外企业以及华为、腾讯、阿里巴巴、百度、浪潮、联想等国内企业

□个人开发者

- ✓ Linus Torvalds, Github上有6500多万的程序员

Apache软件基金会



□组织特点

- ✓ 成立于1999年的非盈利性组织
- ✓ 所管理的开源软件项目都遵循Apache许可证

□组织构成

- ✓ 管理有约2.27亿行代码和350多个开源软件项目
- ✓ 拥有800多名基金会成员，4万多代码贡献者，48万个体

□代表性的开源软件

- ✓ Apache HTTP Server、Derby、Hadoop、Lucene、Tomcat、Ant、Maven等

Google主导的开源软件项目

- ❑机器学习系统 TensorFlow
- ❑网站前端开发工具集MDL
- ❑Web 前端框架 Angular
- ❑基于 Chrome 浏览器的开发环境Spark
- ❑容器集群管理系统 Kubernetes
- ❑Java常用库 Guava、
- ❑Java 编译器 Traceur
- ❑.....



问题思考和讨论

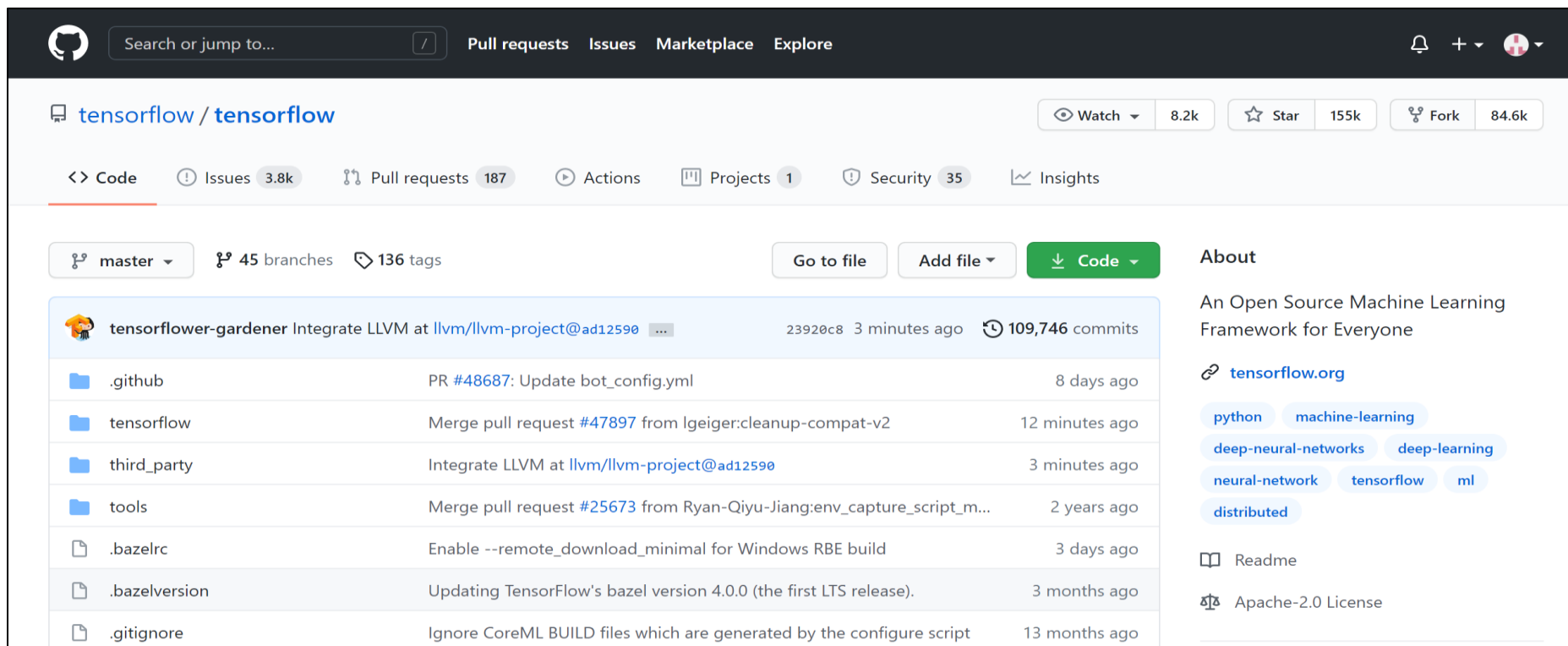
- 开源软件会带来什么样的好处？
- 开源软件是如何开发出来的？
- 为什么那么多的企业和个人关注和参与开源软件？



2.3 开源软件托管平台-Github

□全球最有影响力的开源软件托管平台

□www.github.com



Github为开源软件的托管提供什么样的服务?



□ 一个有影响力的开源软件开发平台和仓库，为开源软件提供存储、协作和发布的支持

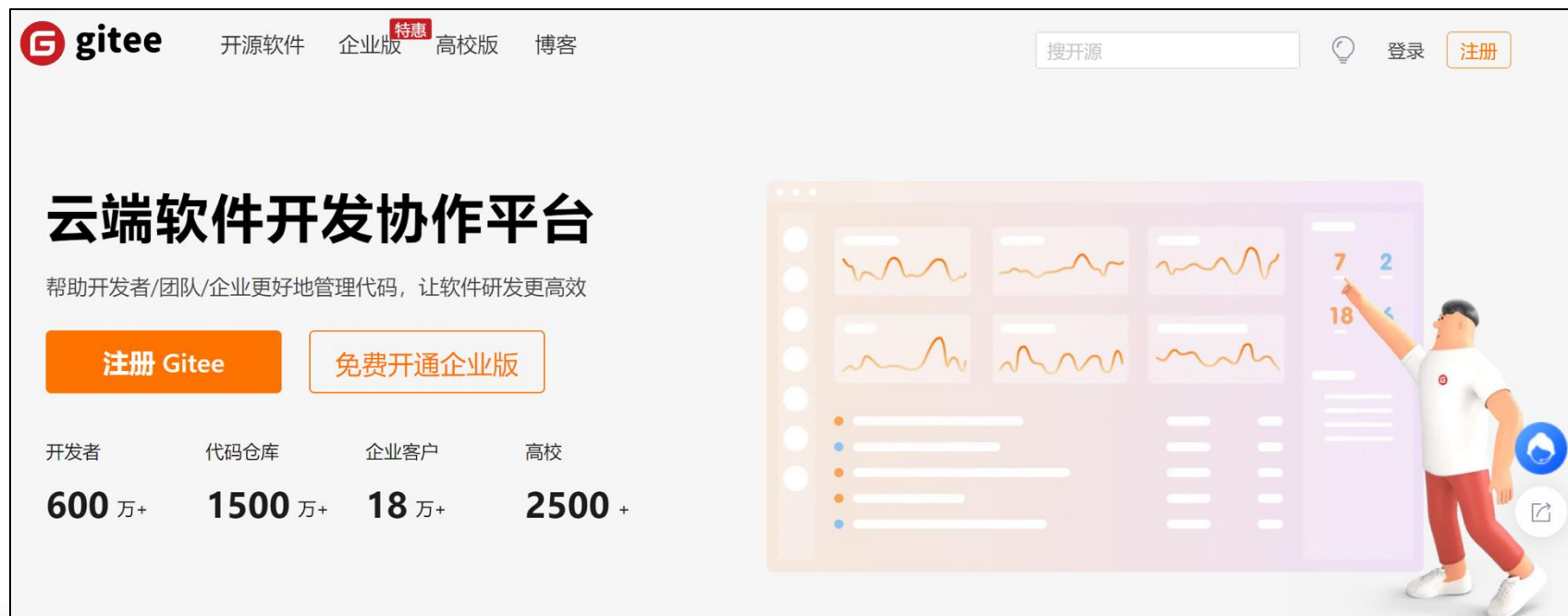
□ sourceforge.net



□中国的开源软件托管平台

✓ 汇聚了国内众多的开发者和主要软件企业的开源软件项目，如华为OpenHarmony操作系统、百度的PaddlePaddle等

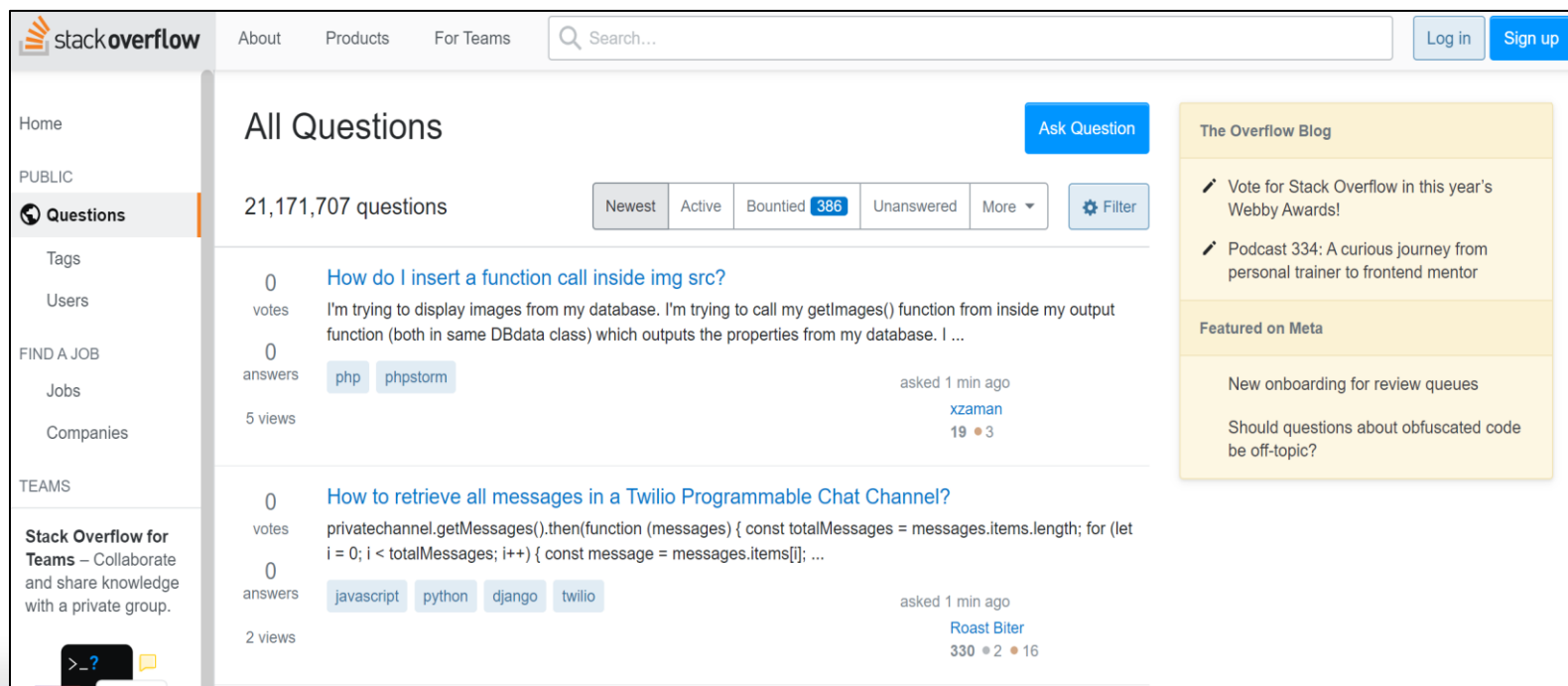
□ www.gitee.com



Stack Overflow

❑ 全球最大的**软件开发知识分享平台**，它支持开发者交流和讨论软件开发问题、分享软件开发经验和知识，进而促进相关开发问题的解决，推动开源软件的开发

❑ **stackoverflow.com**



2.4 开源软件的优势

□采购和开发的成本更低

- ✓开源软件通常是免费的，即使要付费，其费用也非常低廉

□软件质量更高、更安全

- ✓核心代码都在公众的视野之中，代码问题（如缺陷、安全漏洞等）很容易被人发现

□软件研制和交付的更快

- ✓基于开源软件的项目开发可以更为快速地给用户交付软件产品

□软件功能更为强大

- ✓大量的软件开发者不仅参与软件开发，贡献他们的代码，而且还参与软件的创新，提出和构思软件需求，不断完善软件功能

为什么开源软件具有这些优势？



开源软件许可证

□何为开源许可证

- ✓ **声明获得开源代码后拥有的权利，界定对别人的开源作品进行何种操作、何种操作是被禁止的，规范开源软件的使用要求和约束**
- ✓ 开发者虽然可自由地获取开源软件的源代码，但在如何使用开源软件方面还需要遵循相关的开源软件协议

□为什么需要开源许可证

- ✓ 尝试在开源软件的自由创新与创业利益之间达成某种平衡
- ✓ 既支持开发者基于开源软件进行创新，也保护贡献者和创新者的相关利益，同时寻求某些商业运作模式，促进开源软件长期、持续和良性的发展

二类许可证

□宽松式开源许可证

- ✓对用户的限制很少，用户甚至可以将修改后的开源代码后闭源。
- ✓三方面特点：代码使用没有任何限制；用户自担代码质量的风险；用户使用开源软件时须披露原始作者
- ✓BSD、Apache、MIT等都属于宽松式许可证类别。

□Copyleft式开源许可证

- ✓“Copyleft”是Copyright反义词，意指可不经许可随意复制
- ✓更多限制，带有许多条件和要求，比如分发二进制代码时须提供源代码，修改后的开源软件须与之前保持一致的许可证等
- ✓GPL、MPL等属于Copyleft 许可证类别。

Apache开源许可证

□许可证的内容

- ✓该软件及其衍生品须继续使用Apache许可协议
- ✓如果修改了程序源代码，需要在文档中进行声明
- ✓若软件是基于他人源代码编写而成的，则需要保留原始代码的协议、商标、专利声明及其他原作者声明的内容信息
- ✓如果再发布的软件中有声明文件，则需在此文件中标注Apache许可协议及其他许可协议

□Hadoop、Apache HTTP Server、MongoDB等开源软件都基于该许可证。

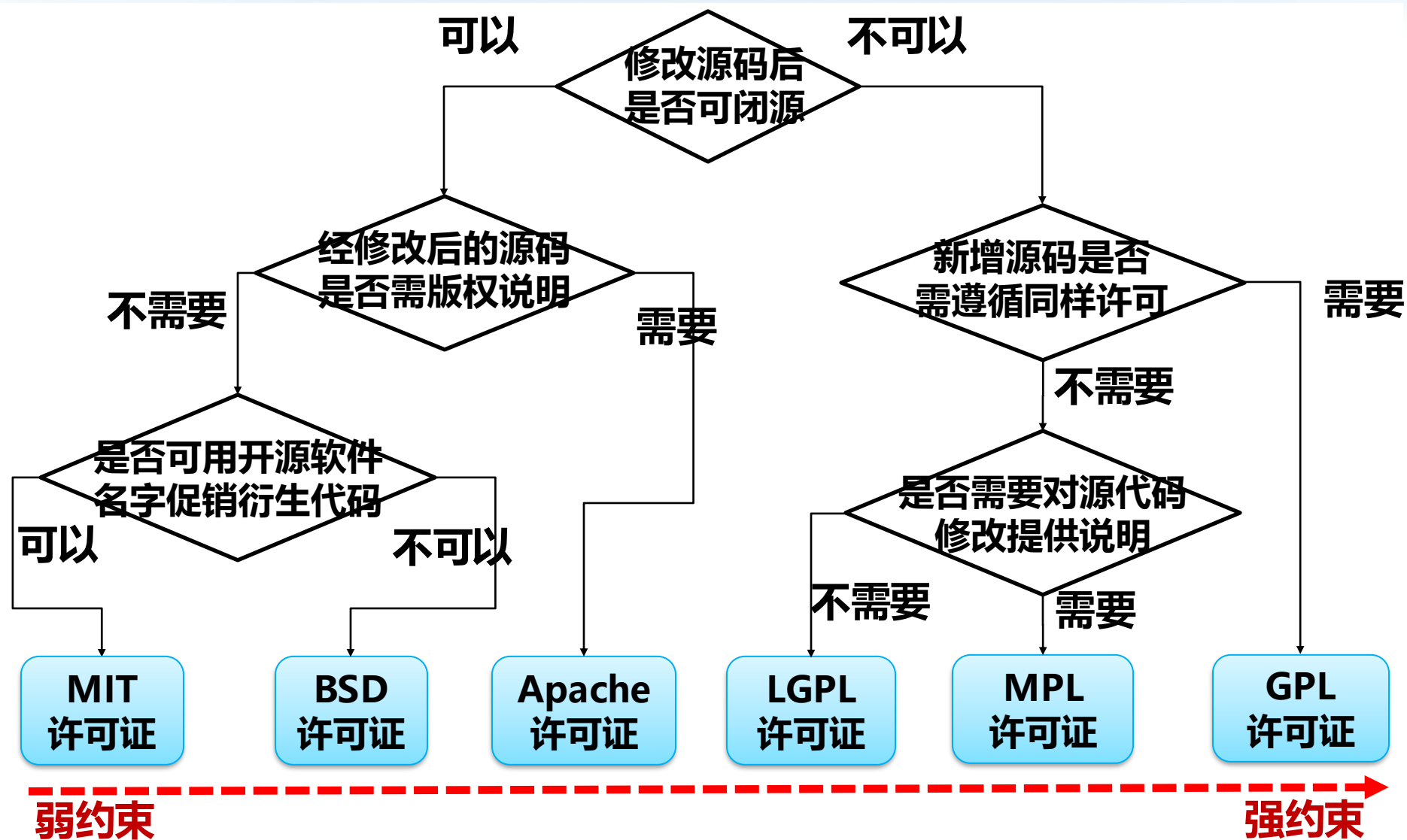
GPL开源许可证

□许可证的内容

- ✓自由复制，对复制的数量和去处不做限制
- ✓自由传播，允许软件以各种形式进行传播
- ✓收费传播，允许出售该软件，但必须让买家知道这个软件是可免费获得的
- ✓修改自由，允许开发者增加或删除软件功能，但修改后的软件必须依然采用GPL许可证

□Linux开源软件采用的就是GPL许可协议

不同许可证的差别



2.5 开源软件的利用

□学习开源软件

- ✓不仅反映了核心开发者的软件技术和功能创意，而且还蕴含了高水平的软件开发技能，如架构设计、编码风格、模块封装等

□重用开源代码

- ✓开源软件构成了支撑软件开发的可重用软件资源库，可实现更大粒度的软件重用

□参与开源建设，掌握开源技术，理解开源文化

- ✓包括反馈软件缺陷、提出软件需求、讨论问题的解决、贡献程序代码等等

开源软件的使用

□重用和集成开源软件来快速搭建高质量的系统

- ✓ Instagram (abbr. Instant telegram) 软件，实现基于移动设备的照片和视频共享、在线服务软件
- ✓ **5个软件工程师**，用了**十多款开源软件**，**8周**打造最初Instagram



**提高软件开发效率
和质量的有效手段**



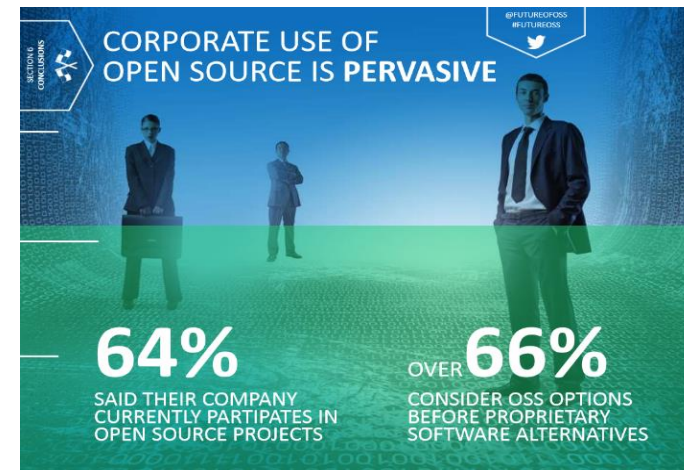
引发软件产业界的转变



78%公司基于开源运行；少于3%公司完全不使用开源



89%的公司认为利用开源大幅度提高了软件创新速度



64%公司参与开源；超过66%公司优先考虑利用开源

- 贡献开源和利用开源已成为工业界的广泛共识和实践，逐渐成为一种重要的软件开发模式 - 《2015开源软件年度调查》
- 当前几乎所有的主要IT企业都在参与开源软件实践，包括微软



内容

1. 软件

✓软件的概念、特点、组成及生命周期

2. 开源软件

✓开源软件实践，托管平台和社区，开源许可证

3. 软件质量

✓软件质量要素及模型

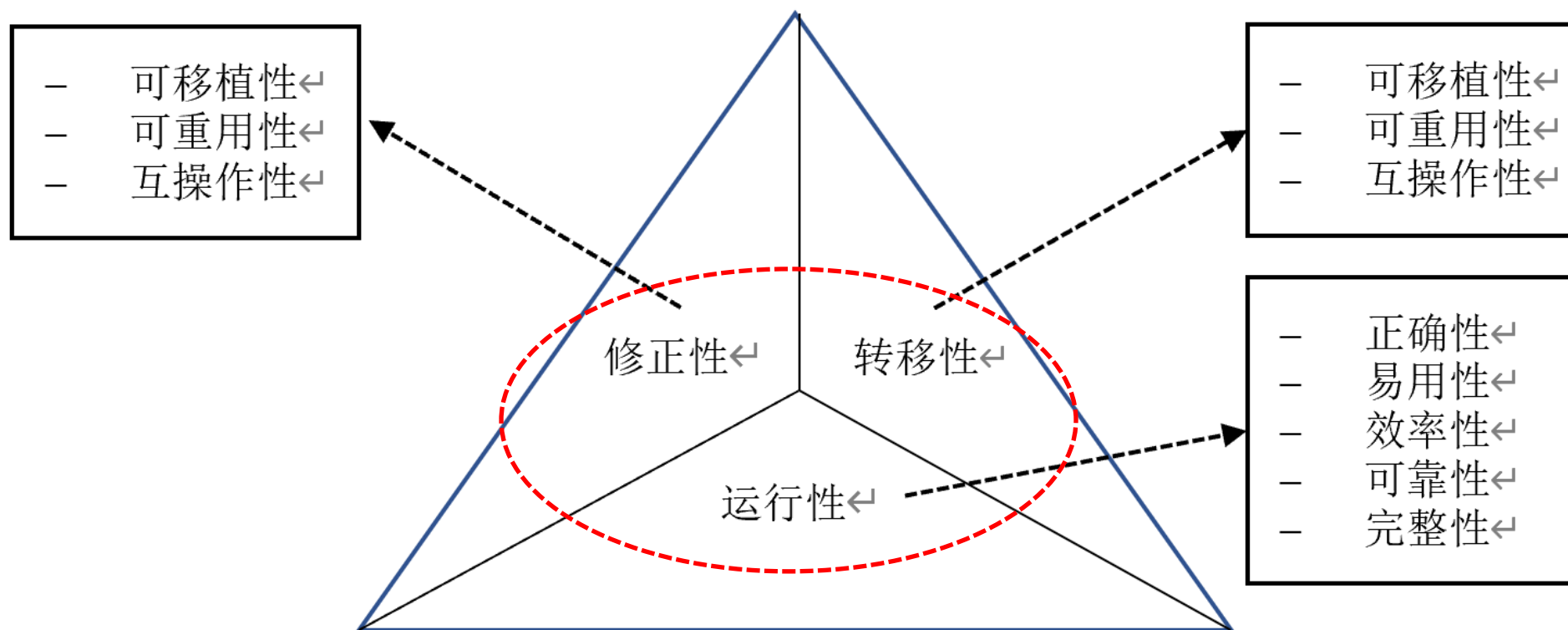
4. 当前软件特征的变化

✓地位、运行环境、形态、复杂性



3.1 软件质量

□软件质量是指软件满足给定需求的程度，它是产品生命线



3.2 软件质量要素 (1/3)

□正确性 (Correctness)

- ✓ 软件满足规格说明和用户要求的程度，即在预定环境下能正确地完成预期功能和非功能需求的程度。

□可靠性 (Reliability)

- ✓ 在规定条件下、限定时间范围内，软件系统完成预期功能、不引起系统故障的能力。

□健壮性 (Robustness)

- ✓ 在计算环境发生故障、输入无效数据或操作错误等意外情况下，软件仍能做出适当响应的程度。

□有效性 (Efficiency)

- ✓ 软件利用计算资源和存储资源以实现其功能的能力。

软件质量要素 (2/3)

□安全性

- ✓包括系统安全 (Safety) 和信息安全 (Security) , 系统安全是指软件能及时有效地避免给人员、设施、环境、经济等造成损害, 信息安全是指软件能有效防控各类的非法获取、传播和使用

□可维护性 (Maintainability)

- ✓是否易于对其进行修改以更正错误、增强功能、适应新运行环境

□可移植性 (Portability)

- ✓把软件从一种运行环境转移到另一种运行环境下运行的难易程度

□可重用性 (Reusability)

- ✓软件模块、构件、设计方案等在其他软件开发中被再次使用程度

软件质量要素 (3/3)

□可理解性 (Comprehensibility)

- ✓ 软件开发者或用户理解该软件系统的容易程度

□可信性 (Dependability)

- ✓ 采取有效措施确认软件满足人们的要求和期望

□持续性 (Sustainability)

- ✓ 面对各种突发异常事件，仍能提供令人满意的的服务的能力

□可用性 (Usability)

- ✓ 使用和操作软件系统难以程度

□互操作性 (Interoperability)

- ✓ 软件系统与其他系统进行交换信息、协同工作的能力

2019年华为公司发布的第一号文件

□ “全面提升软件工程能力与实践，打造可信的高质量产品”

✓把可信作为第一优先级，放在功能、特性和进度之上

□可信软件的基本要求

- ✓**安全性**以抗击攻击，保护业务和数据的机密性、完整性和可用性
- ✓**韧性**（Resilience）以确保系统受攻击时仍能保持良好的运行状态，具有快速恢复的能力
- ✓**隐私性**（Privacy）以保护用户的数据及其隐私
- ✓**可靠性**和可用性以确保软件产品长期无故障运行，具备快速恢复和自我管理的能力，提供可预期的、一致的服务。

内容

1. 软件

✓软件的概念、特点、组成及生命周期

2. 开源软件

✓开源软件实践，托管平台和社区，开源许可证

3. 软件质量

✓软件质量要素及模型

4. 当前软件特征的变化

✓地位、运行环境、形态、复杂性

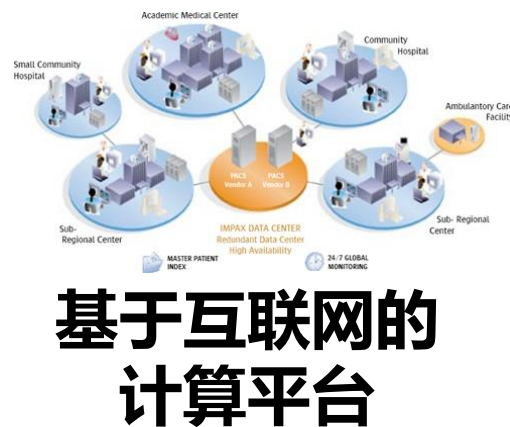
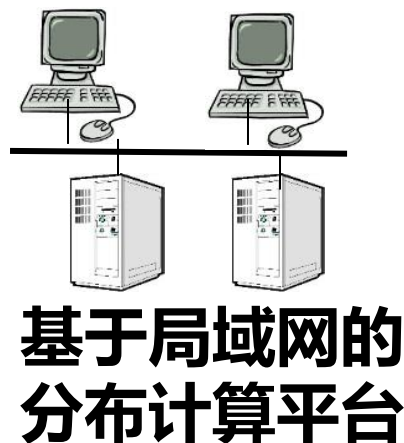
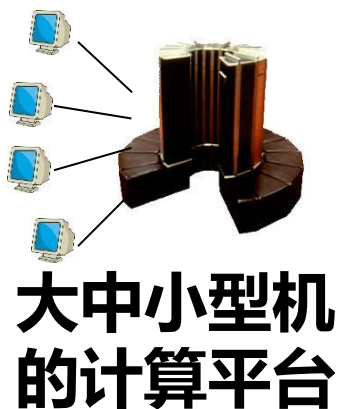


4.1 软件的地位和作用

- 深入到社会、经济、生活的方方面面，**无处不在**
- 作为一种**创新的工具**，深刻地改变各个行业和领域，创新问题解决模式和方式，影响人们日常学习、生活和工作
 - ✓ “12306” 改变了购票方式，淘宝和京东改变了购物方式
- 软件是实现行业和领域进行信息化融合和改造，实现创新性发展的**使能技术和重要利器**
 - ✓ 企业信息化改造，图书借阅等
- 软件已经成为人类社会的**关键性基础设施**
 - ✓ 就像电力、交通设施一样，发挥基础性的作用

4.2 软件的运行环境

从前端的PC终端、可穿戴设备、智能手机到后端的云中心、高性能计算中心



无处不在的
计算

从孤立、独立、局域和可控的计算环境到分布、开放、动态、难控、无处不在计算环境

4.3 软件形态

□不是单一系统，而是**系统之系统**

✓诸多系统联盟和组合而成

□不仅是社会技术系统，还是一种**生态系统**

✓社会、信息、物理等要素共存

□不是同构系统，而是异构、多样的**系统联盟**

✓要素异构，客观存在，也是必然

□不在封闭环境，而在**开放环境中**

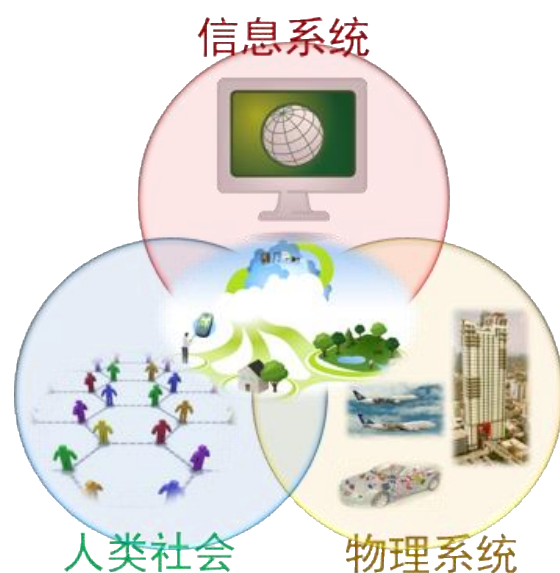
✓环境和系统相互作用，环境开放如互联网、战场环境

□不是封闭系统，而是**动态适应系统**

✓要素、关系、联盟等持续变化，边界不明确

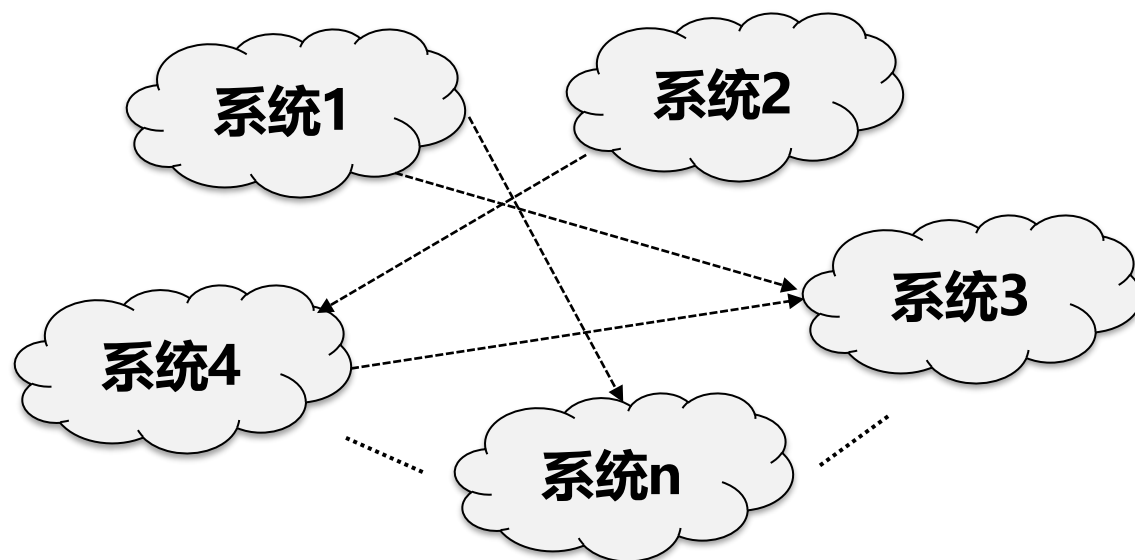
软件形态-社会技术系统

- 由人、社会组织、物理设备、过程等要素共同组成和相互作用的人机物共生系统
- 软件系统无法单独存在，需与物理、社会系统交互作用



软件形态-系统子系统

- 由一组面向任务、服务于不同对象的**子系统**构成
- 每个子系统可**独立运作**并能提供相对独立功能
- 整个系统通过各独立系统间**交互**来实现全局任务



软件形态-分布式异构系统

- 拥有大量形式多样、地理或者逻辑上是分布的，分散部署在互联网上的软件实体
- 软件实体的分布性是必须的，因为越来越多的应用本身就是分布的，软件实体的分布性有助于提高软件系统的可靠性和安全性
- 构成软件系统的软件实体通常是异构的，异构性是一种必然

软件形态-动态演化系统

- 系统的边界和需求的不确定性和持续演变性
- 动态开放的特点，软件系统需要根据外部环境的变化而不断地调整自身，包括系统的体系结构和交互协作等等，进而表现出持续演化的特点
- 系统的运维和系统的运行需要交织在一起

城市交通管理系统
银行服务信息系统
医疗保证信息系统
作战指挥控制系统

.....

软件形态-系统联盟

□大规模复杂信息技术系统是由大量相对独立、自我控制和管理系统组装而成—系统联盟

- ✓分属不同的组织，并由他们来管理
- ✓共同工作和相互共存
- ✓并非一开始就设计好，而是演化而成
- ✓联盟动态变化

开发系统联盟需要采用社会技术观点，借助系统工程的方法

DOI:10.1145/2200249.2200289

The reductionism behind today's software-engineering methods breaks down in the face of systems complexity.

BY IAN SOMMERVILLE, DAVE CLIFF, RADU CALINESCU, JUSTIN KEEN, TIM KELLY, MARTA KWIATKOWSKA, JOHN MCDERMID, AND RICHARD PAIGE

Large-Scale Complex IT Systems

ON THE AFTERNOON of May 6, 2010, the U.S. equity markets experienced an extraordinary upheaval. Over approximately 10 minutes, the Dow Jones Industrial Average dropped more than 600 points, representing the disappearance of approximately \$800 billion of market value. The share price of several blue-chip

multinational companies fluctuated dramatically; shares that had been at tens of dollars plummeted to a penny in some cases and rocketed to values over \$100,000 per share in others. As suddenly as this market downturn oc-

curred, it reversed, so over the next few minutes most of the loss was recovered and share prices returned to levels close to what they had been before the crash.

This event came to be known as the "Flash Crash," and, in the inquiry report published six months later,² the trigger event was identified as a single block sale of \$4.1 billion of futures contracts executed with uncommon urgency on behalf of a fund-management company. That sale began a complex pattern of interactions between the high-frequency algorithmic trading systems (algorithms) that buy and sell blocks of financial instruments on incredibly short timescales.

A software bug did not cause the Flash Crash; rather, the interactions of independently managed software systems created conditions unforeseen

» key insights

- Coalitions of systems, in which the system elements are managed and owned independently, pose challenging new problems for systems engineering.
- When the fundamental basis of engineering—reductionism—breaks down, incremental improvements to current engineering techniques are unable to address the challenges of developing, integrating, and deploying large-scale complex IT systems.
- Developing complex systems requires a socio-technical perspective involving human, organizational, social, and political factors, as well as technical factors.

软件形态-生态系统

□“运行”生态

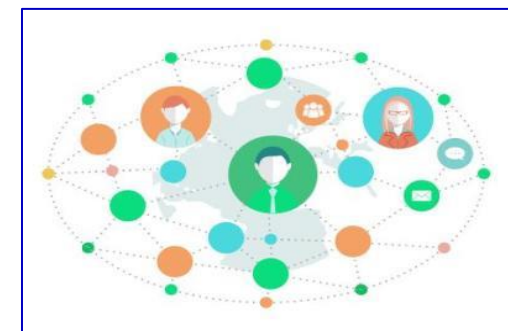
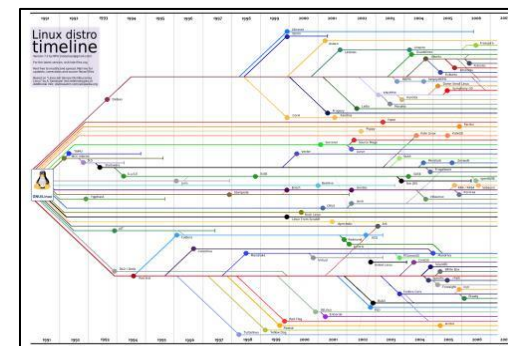
- ✓软件的运行需要依靠“生态”的支撑
- ✓从计算和物理设备、社会系统到OS和运行环境

□“演化”生态

- ✓软件的演化基于特定的“生态”支撑
- ✓从最初的“源头”到最新“版本”

□“人员”生态

- ✓涉及到大量、开放的介入人员
- ✓使用者、开发者、运维者等



生态：共同环境 + 诸多要素 + 独立演化 + 相互依存

4.4 软件系统的规模

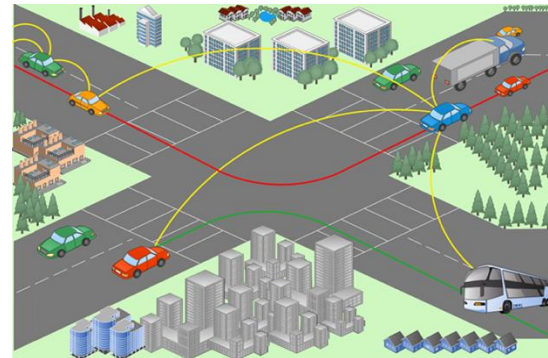
□构成软件系统的代码行数量、软件运行时的进程和线程以及它们之间的交互数量、软件需处理的数据量、软件连接各类设备和人员数量等不断增加

□示例

- ✓ Windows XP有3500万行代码量，Windows Vista有5000万行代码量，Windows 7的代码量大约是7000万行
- ✓ 宝马7系的软件总量超过了2亿行，特斯拉S的内嵌软件代码行总量超过4亿行，空客A380中的软件代码行总量超过了10亿行
- ✓ F-35中80%左右的系统功能由软件来实现，软件代码量约为800万行

软件系统的应用及示例

- 高性能计算
- 信息物理系统
- 智能机器人
- 云计算
- 健康医疗
- 城市交通
- 军事信息系统
- 航空航天



4.5 我国软件建设的挑战和使命

□现实挑战：受制于人

- ✓ 微软Windows垄断了桌面OS
- ✓ Google垄断了智能终端（如手机、平板电脑）的OS
- ✓ 工业软件和科学计算软件等领域，差距更大

□整体实力不足

- ✓ 缺乏在国际上有影响力的软件企业 and 产品

□需要加大、加快国产化软件系统的研发和应用

- ✓ 使命伟大，责任重大，任重而道远

小结

□软件

- ✓由程序 + 文档 + 数据组成

□软件特点

- ✓逻辑产品、设计开发、需求易变、系统复杂、缺陷隐蔽

□开源软件

- ✓代码可自由获取和传播，需遵循许可证，充分利用开源软件

□软件质量

- ✓多要素，如正确性、可靠性、可信性等

□软件发生的变化

- ✓地位和作用，形态和复杂性，运行环境，系统规模

综合实践一

□任务：选取或指定待阅读、分析和维护的开源软件。

□方法

- ✓访问Github、码云Gitee、SourceForge等开源软件托管平台，从中检索到符合上述要求的开源软件，下载或克隆开源软件代码，阅读开源软件的相关文档来安装、部署和运行开源软件。以二人为一组、采用结对方式来开展本综合实践

□要求

- ✓所选取或指定的开源软件要求功能易于理解、代码质量高、规模适中（5000-20000行代码量），也可以直接指定“MiNote”便签管理开源软件作为阅读、分析和维护的对象。

□结果：获得开源软件源代码，并可运行和操作该开源软件

□任务：查看和分析开源软件

□方法

- ✓访问Github、SourceForge、Gitee等开源软件托管平台或Apache、Eclipse等开源软件基金会平台，从中检索自己感兴趣的开源软件，阅读相关的软件文档，下载安装开源软件。

□要求

- ✓结合自己的兴趣，查看有那些开源软件，分析这些软件的功能和定位、存在的缺陷和不足。

□结果

- ✓掌握开源软件托管平台使用方法，了解感兴趣的开源软件情况。

思考和讨论

与硬件系统（如手机、电视机）相比较哦，软件系统有何特殊性？它们给软件系统的开发和维护带来什么样的新挑战？



问题和讨论

