



第一章 软件工程概述

目录 >>>

一

软件工程学科概览

二

IT行业人才格局及成长路线

三

软件过程—软件工程的核心组成部分

四

本课程的要点说明

目录 >>>

一

软件工程学科概览

二

IT行业人才格局及成长路线

三

软件过程—软件工程的核心组成部分

四

本课程的要点说明

认识软件工程 >>>

思考

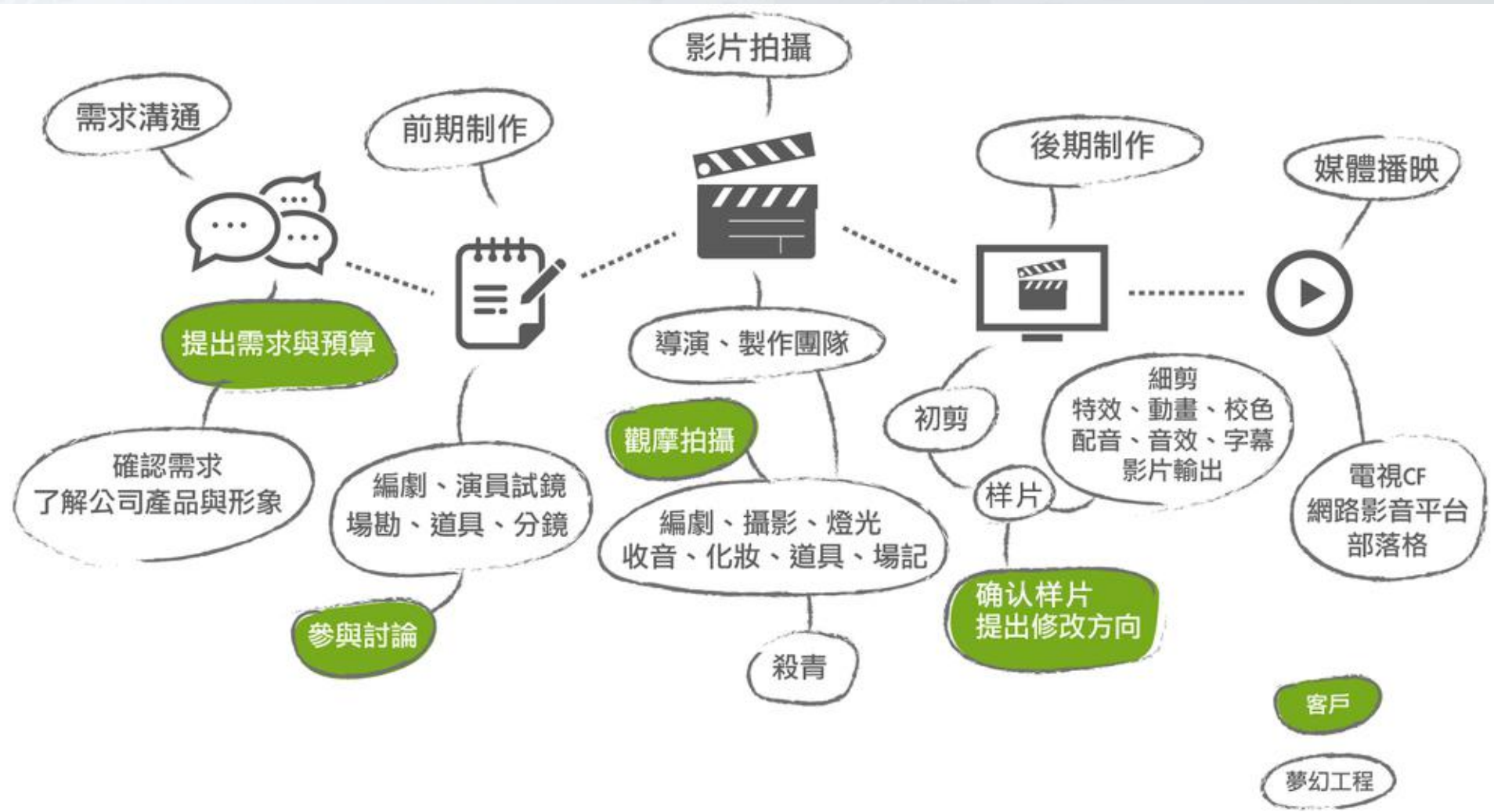
- 掌握一项软件开发技术就能很好就业，为啥还要学软件工程？
- 软件项目的成功和软件工程学有什么关系？
- 软件工程专业究竟“长的什么样子”？



演員是不是電影的全部？



Ellen DeGeneres
@TheEllenShow
If only Bradley's arm was longer.
11:06 AM - 3 Mar 2014



同样，软件项目也不只有开发人员，
不只是开发环节

开发人员很关键，其它岗位同样关键；
能够组织、管理、遵循流程主导整个
项目更关键



软件项目并不是个个都成功 >>>

- Standish Group公司的调查结果 [365家公司的8380个项目]

| MODERN RESOLUTION FOR ALL PROJECTS | | | | | |
|------------------------------------|------|------|------|------|------|
| | 2011 | 2012 | 2013 | 2014 | 2015 |
| SUCCESSFUL | 29% | 27% | 31% | 28% | 29% |
| CHALLENGED | 49% | 56% | 50% | 55% | 52% |
| FAILED | 22% | 17% | 19% | 17% | 19% |

报告参见：<https://www.infoq.com/articles/standish-chaos-2015/>

什么是决定软件项目成功的关键因素？>>>

| 成功因素 | 权重 |
|--------------|-----|
| 管理层的支持 | 15% |
| 用户的参与 | 15% |
| 团队综合战斗力 | 15% |
| 最佳实践 | 15% |
| 有才能的员工（商务+技术 | 10% |
| 规范的管理架构 | 8% |
| 采用敏捷过程 | 7% |
| 合适的规划与执行 | 6% |
| 专业的项目管理 | 5% |
| 清晰的商业目标 | 4% |



软件工程学的存在价值：促进软件项目成功 >>>



软件的概念 >>>

1、软件

---- software

soft + ware

软制品(软体)

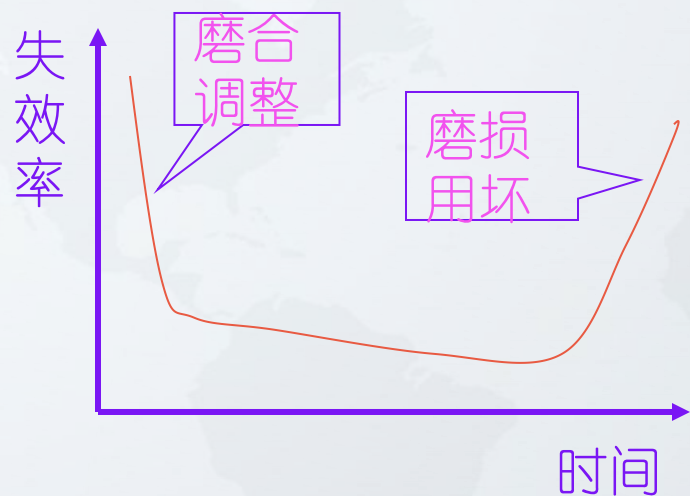
软件是计算机系统中与硬件相互依存的另一部分。它包括程序、数据及其相关文档的完整集合。

- (1) 能够完成预定功能和性能的可执行指令 (program)
- (2) 使得程序能够适当地操作信息的数据结构 (data)
- (3) 描述程序的操作和使用的文档 (document)

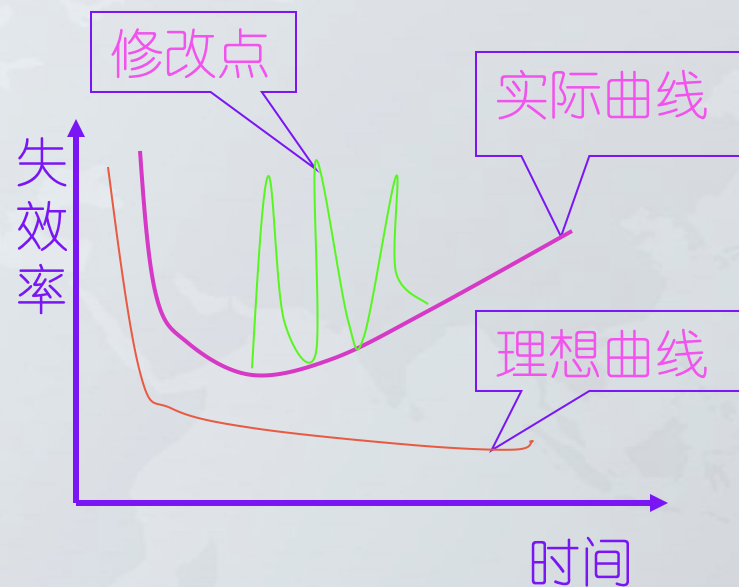


2、软件特点

- 软件是一种逻辑实体，而不是具体的物理实体。
- 软件的生产与硬件不同。
- 在软件的运行和使用期间，没有硬件那样的机械磨损，老化问题，但它存在退化问题，开发人员必须维护软件。
- 大多数软件是自定的，而不是通过已有构件组装而成的。
- 软件成本相当昂贵。
- 软件本身是复杂的。



硬件失效率曲线



软件失效率曲线



3、软件危机

软件危机定义：软件在开发和维护过程中遇到的一系列严重问题。

软件危机包含两层含义：

如何开发软件。

如何维护数量不断膨胀的已有软件。



软件危机的表现：

- (1) 软件开发的**进度难以控制**，经常出现经费超预算、完成期限拖延的现象。
- (2) 软件**需求在开发初期不明确**，导致矛盾在后期集中暴露，从而对整个开发过程带来灾难性的后果。
- (3) 软件**文档资料不完整、不合格**。由于缺乏完整规范的资料，加之软件测试不充分，从而造成软件质量低下。
- (4) 软件的**可维护性差**，程序错误难以改正，程序不能适应硬件环境的改变。
- (5) 软件**价格昂贵**，软件成本在计算机系统总成本中所占的比例逐年上升。



软件危机的原因：

- （1）**客户**对软件需求的描述不精确，可能有遗漏、有二义性、有错误，在软件开发过程中，用户提出修改软件功能、界面、支撑环境等方面的要求。
- （2）**软件开发人员**对用户需求的理解与用户的本来愿望有差异。不能有效地、独立自主地处理大型软件的全部关系和各个分支，因此容易产生疏漏和错误。
- （3）**管理人员**、软件开发人员等各类人员的信息交流不及时、不准确、有时还会产生误解。
- （4）缺乏有力的**方法和工具**方面的支持，过分地依靠程序人员在软件开发过程中的技巧和创造性，加剧软件产品的个性化。



克服危机的途径：

1968年秋季，NATO（北约）的科技委员会召集了近50名一流的编程人员、计算机科学家和工业界巨头，讨论和制定摆脱“软件危机”的对策。由于认识到软件的设计、实现、维护和传统的工程规则有相同的基础，在那次会议上**首次提出了“软件工程”**（software engineering）这个概念。

软件工程学的存在价值：促进软件项目成功



1、 “软件工程” ----Software Engineering

是研究和应用如何以系统化的、规范的、可度量的方法去开发、运行和维护软件，即把工程化应用到软件上。

软件工程的历史与发展趋势 >>>

- 1940s，第一台数字计算机诞生、软件、硬件概念诞生；
- 1968年，“软件工程”这个术语第一次使用，作为一个会议标题，该项目由北约（NATO）赞助；该会议确认了要用定义最佳实践的方式帮助改善软件开发；
- 1972年，大卫·帕纳斯提出模块化和隐藏信息的关键概念，以帮助程序员应付日益增加软件系统的复杂性；
- 1981年，软件工程经济学诞生，Barry W. Boehm在他的书中提出了成本模型COCOMO；
- 2000年以后，随着软件需求更多的出现在更小的组织，业界呼唤更廉价的软件解决方案，这导致了更多轻量的软件方法纷纷诞生，Agile就产生于这个背景下；
- 2012年，纪律敏捷交付框架（Disciplined Agile Delivery Framework）发布，这是一个混合框架，采用和扩展了Unified Process、Scrum、XP和其他软件过程；

软件工程学研究的目标



- 软件开发成本较低；
- 软件功能能够满足用户的需求；
- 软件性能较好；
- 软件可靠性高；
- 软件易于使用、维护和移植；
- 能按时完成开发任务，并及时交付使用。

采用先进的软件工程方法，使质量、成本和生产率三者之间的关系达到最优的平衡状态。



软件生存周期：

是指软件产品从考虑其概念开始到该软件产品交付使用，直至最终退役为止的整个过程。一般包括计划、分析、设计、实现、测试、集成、交付、维护等阶段。

注意：

在实践中，软件开发并不总是按照计划、分析、设计、实现、测试、集成、交付、维护等顺序来执行的，即各个阶段是可以重叠交叉的。整个开发周期经常不是明显地划分为这些阶段，而是分析、设计、实现、再分析、再设计、再实现等迭代执行。

软件生存周期的各个阶段主要任务 >>>

1.计划阶段

确定待开发系统的**总体目标**和**范围**。

研究系统的**可行性**和可能的解决方案，对资源、成本及进度进行合理的估算。

2.分析阶段

分析、整理和提炼所收集到的**用户需求**，建立完整的分析模型，将其编写成**软件需求规格说明**和初步的用户手册。

3.设计阶段（总体设计和详细设计）

设计阶段的目标是**决定软件怎么做**。

软件设计主要集中于**软件体系结构、数据结构、用户界面和算法**等方面。

软件生存周期的各个阶段主要任务 >>>

4.实现阶段（编码）

实现阶段是将所设计的各个模块编写成计算机可接受的程序代码。

5.测试阶段

设计测试用例，对软件进行测试，发现错误，进行改正。

6.运行和维护阶段

应当在软件的设计和实现阶段充分考虑软件的可维护性。

维护阶段需要测试是否正确地实现了所要求的修改，并保证在产品的修改过程中，没有做其他无关的改动。

维护常常是软件生命周期中最具挑战性的一个阶段，其费用是相当昂贵的。

软件工程学的肖像

ACM和IEEE-CS发布的SWEBOK定义了软件工程学科的内涵，它由10个知识域构成。

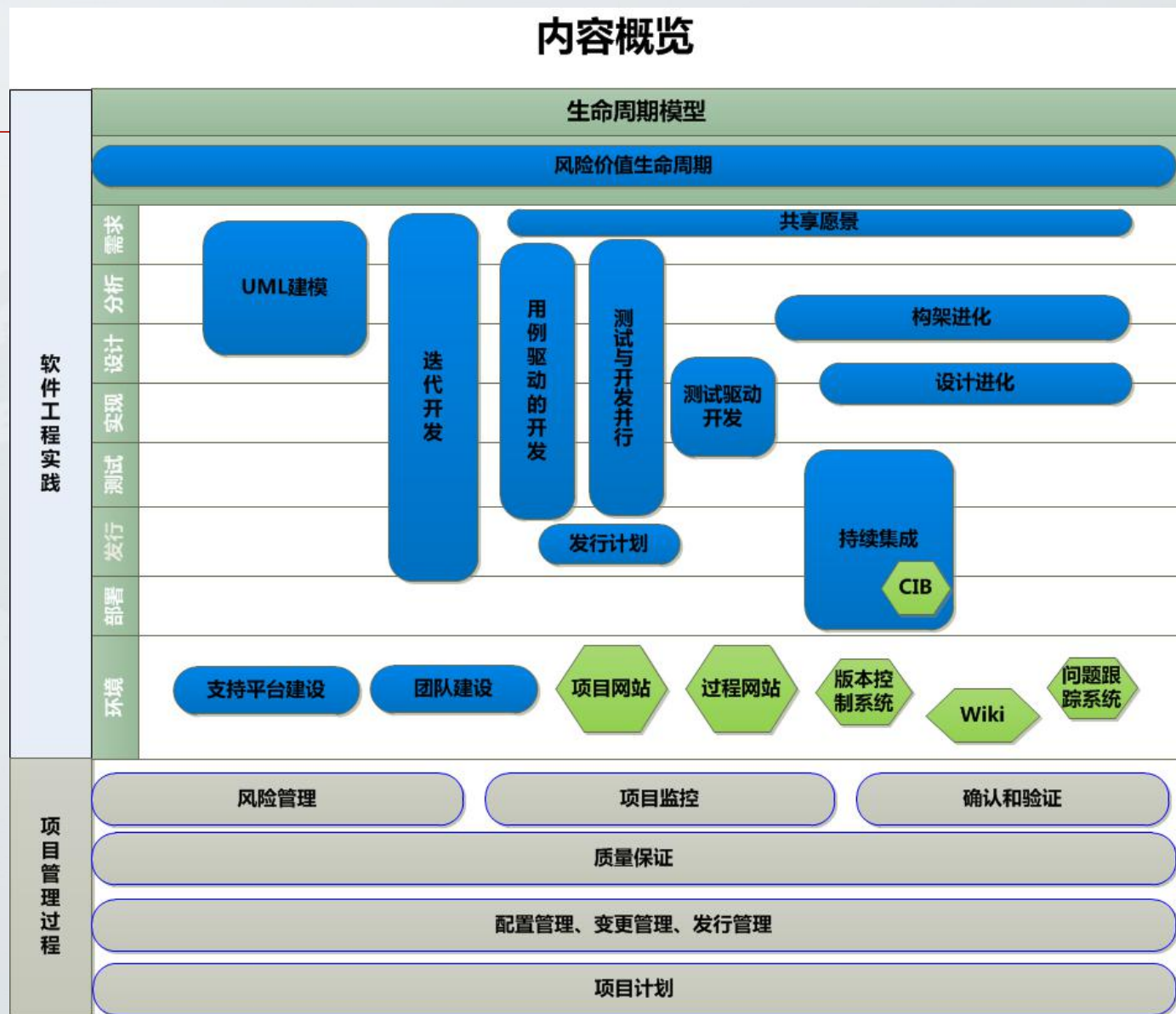
- | | |
|---------------|------------|
| (1) 软件需求 | (2) 软件设计 |
| (3) 软件构造 | (4) 软件测试 |
| (5) 软件维护 | (6) 软件配置管理 |
| (7) 软件工程管理 | (8) 软件工程过程 |
| (9) 软件工程工具和方法 | (10) 软件质量 |

这是我们熟知的软件开发技术

这是我们熟知的软件测试技术

其余的都是软件工程专业人士同样应该熟知的

是不是有种“刚上路的错觉”



目录 >>>

一

软件工程学科概览

二

IT行业人才格局及成长路线

三

软件过程—软件工程的核心组成部分

四

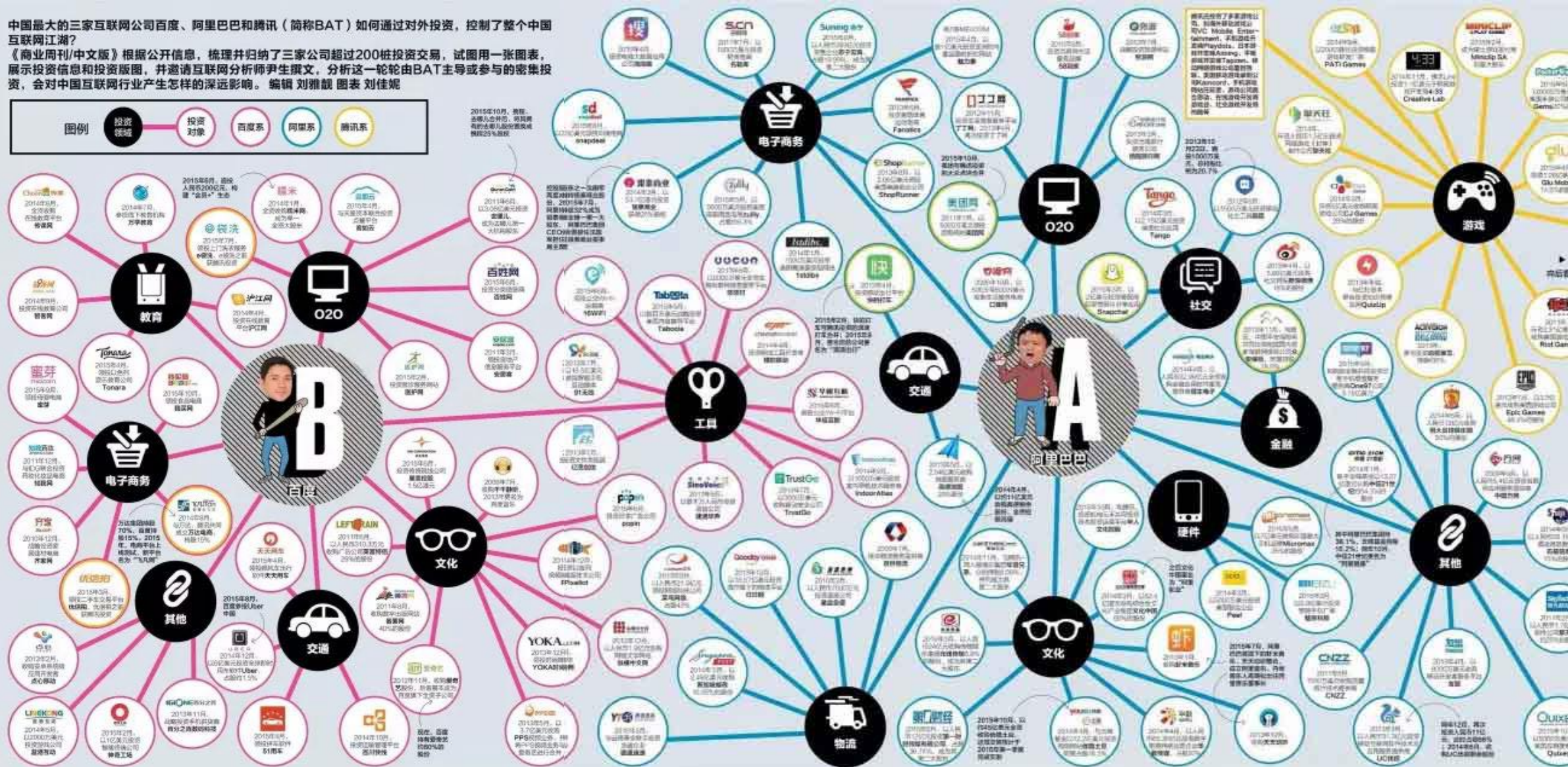
本课程的要点说明

BAT完全霸占互联网江湖

看看今天的IT行业版图



中国最大的三家互联网公司百度、阿里巴巴和腾讯（简称BAT）如何通过对外投资，控制了整个中国互联网江湖？《商业周刊/中文版》根据公开信息，梳理并归纳了三家公司超过200桩投资交易，试图用一张图表，展示投资信息和投资版图，并邀请互联网分析师尹生撰文，分析这一轮轮由BAT主导或参与的密集投资，会对中国互联网行业产生怎样的深远影响。编辑 刘雅靓 图表 刘佳妮



BAT的企业基因？>>>

技术线



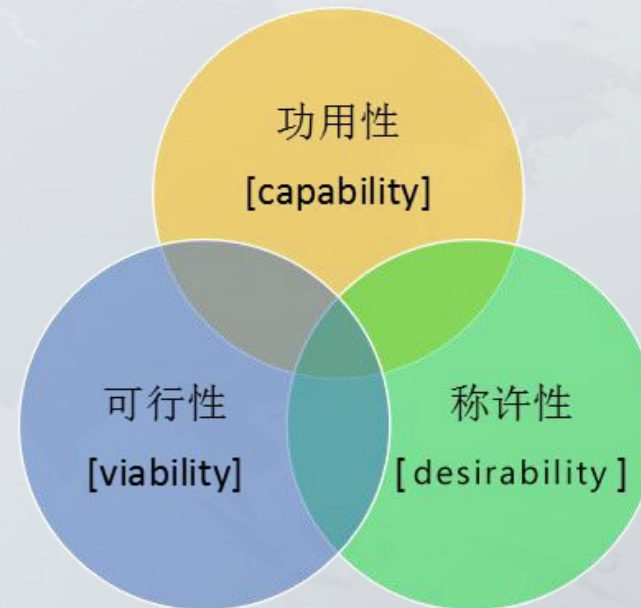
产品线



商业线



你会选择谁？

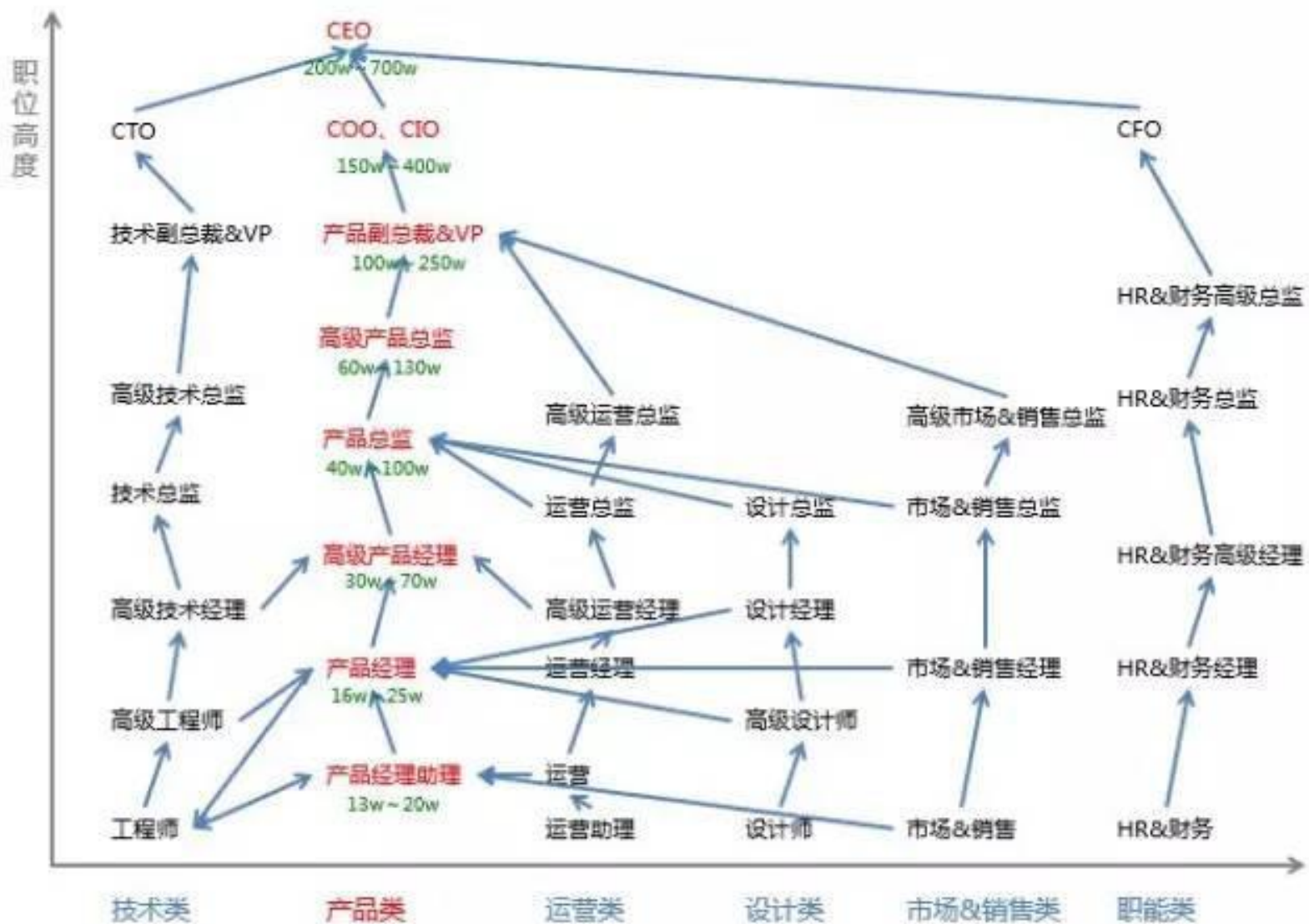


Doblin Group 公司总裁Larry Keely

现代产品开发三原则

产品、技术、商业三位一体

互联网公司的人才格局及成长路线



传统IT公司也有自己的天下 >>>

项目开发

行业软件

软件外包

工具软件

系统软件



传统IT公司的人才格局 >>>



传统IT人才成长路径 >>>

你在学校至少精通了一门编程原因，平时很喜欢写代码，并且编写了大量代码。

你不喜欢编程，一见到代码就头晕，但你喜欢钻研某种技术，如：数据库技术、网络技术等。

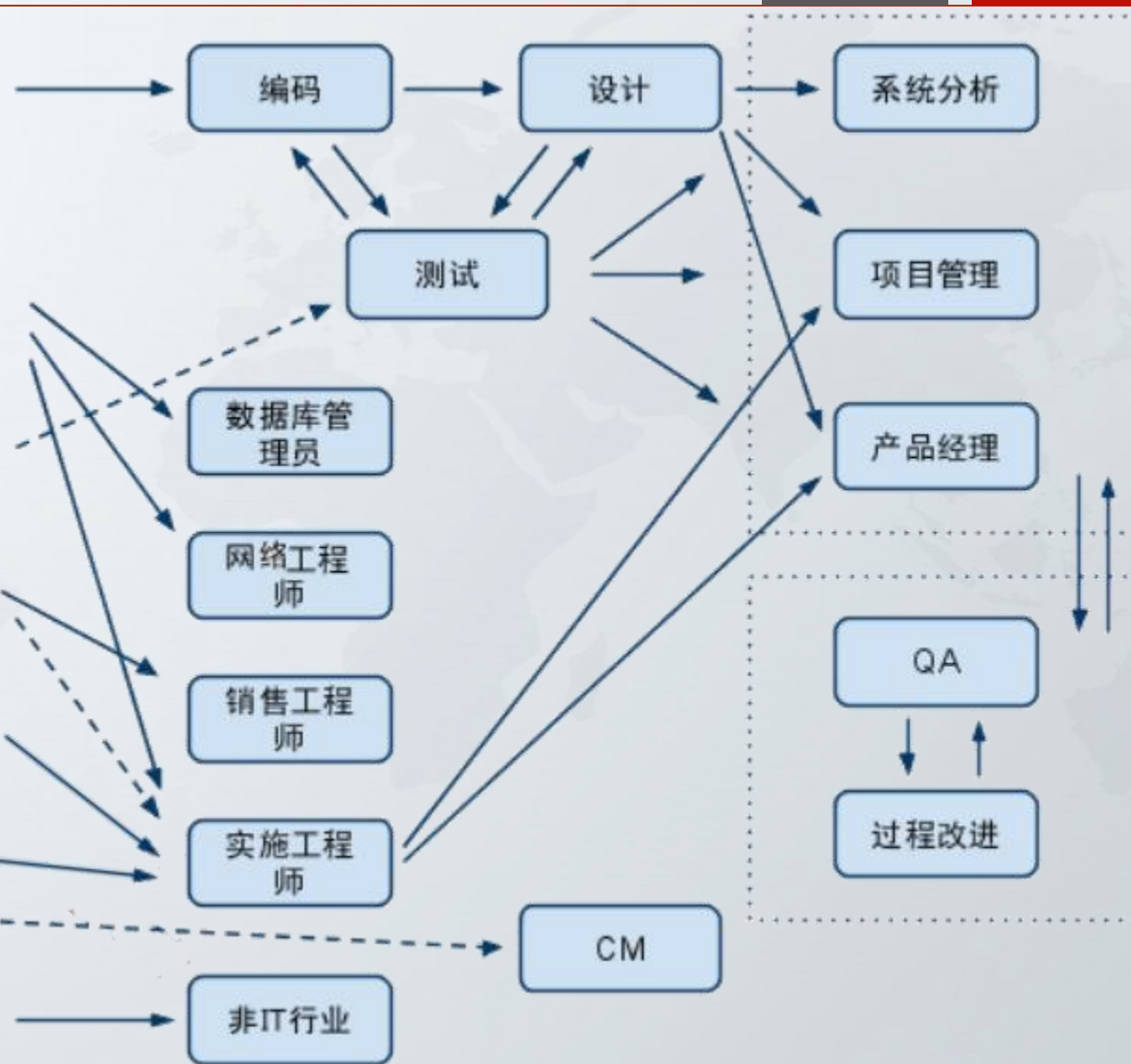
你不太喜欢编程，对技术有一定了解，平时使用软件喜欢找碴。

你比较外向，喜欢交际，能“吹水”。

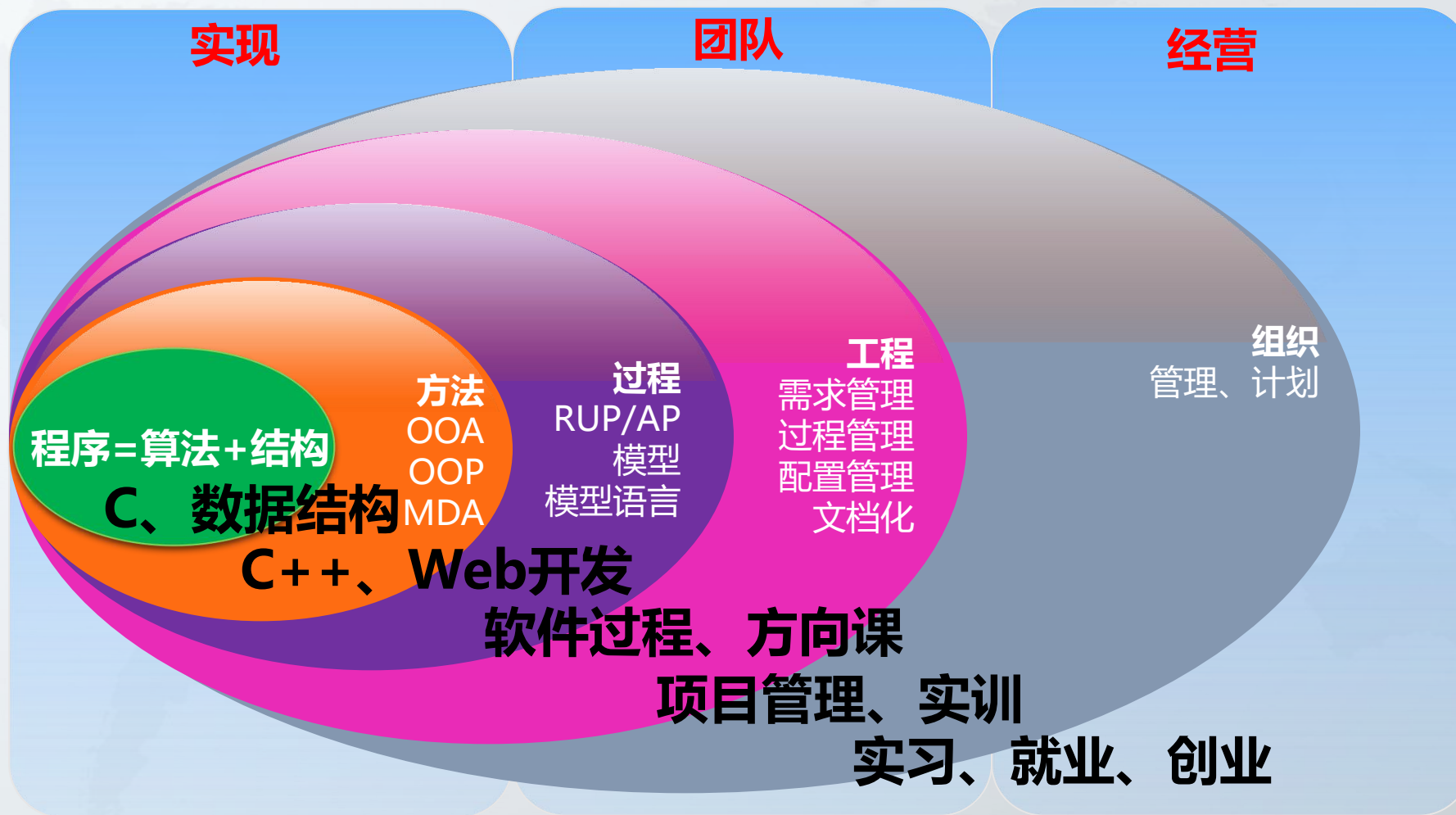
你编码和技术都不喜欢，喜欢软件设计、项目管理。

编码和技术你都不喜欢，你喜欢安逸的工作，想做支持类的工作，如QA、CM。

你在学校几年都不知道学了啥，不知道自己想干什么。



软件学院人才培养架构 >>>



参考自周爱民《大道至简：软件工程实践者的思想》

目录 >>>

一

软件工程学科概览

二

IT行业人才格局及成长路线

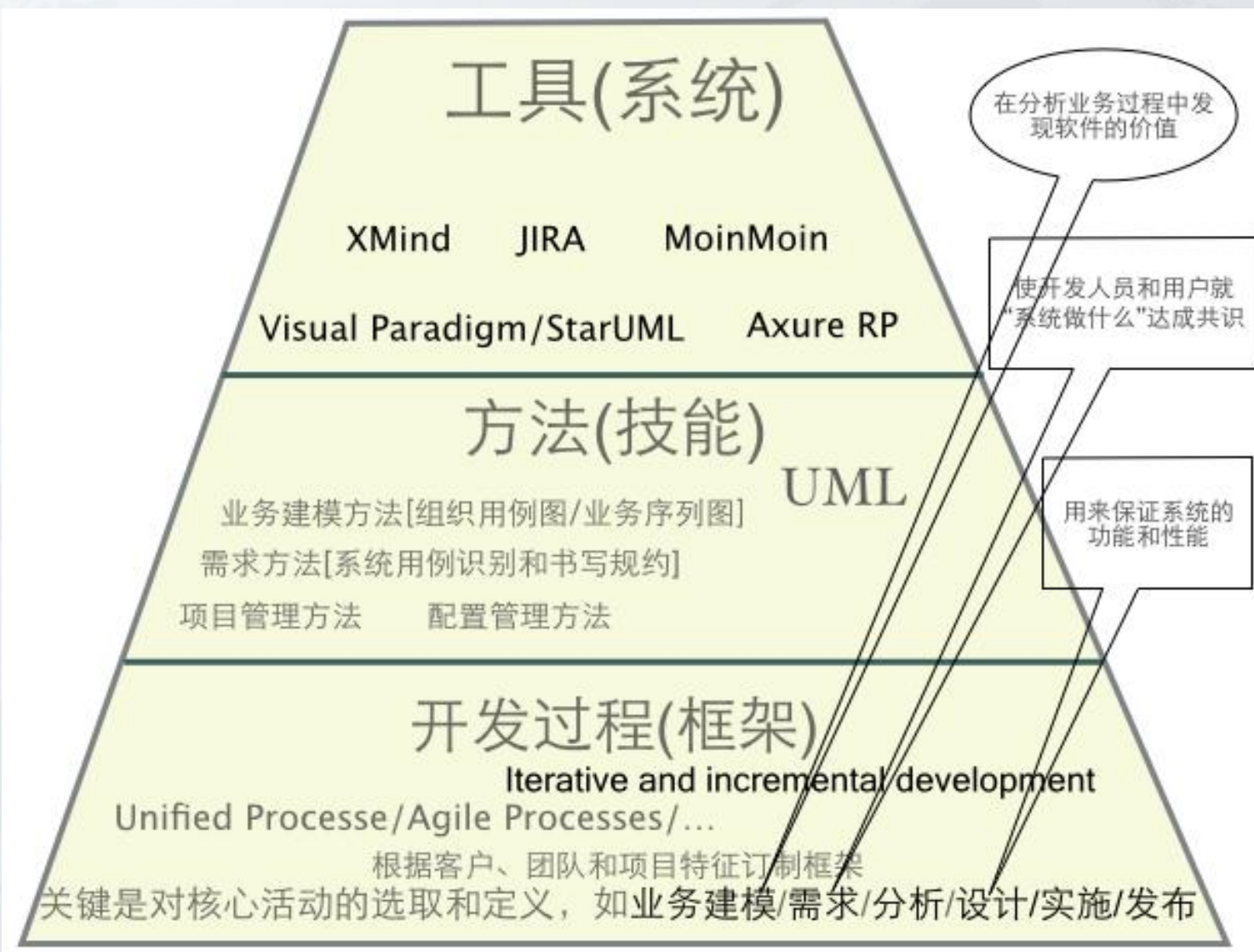
三

软件过程—软件工程的核心组成部分

四

本课程的要点说明

软件工程三要素



瀑布模型 (Waterfall Model)



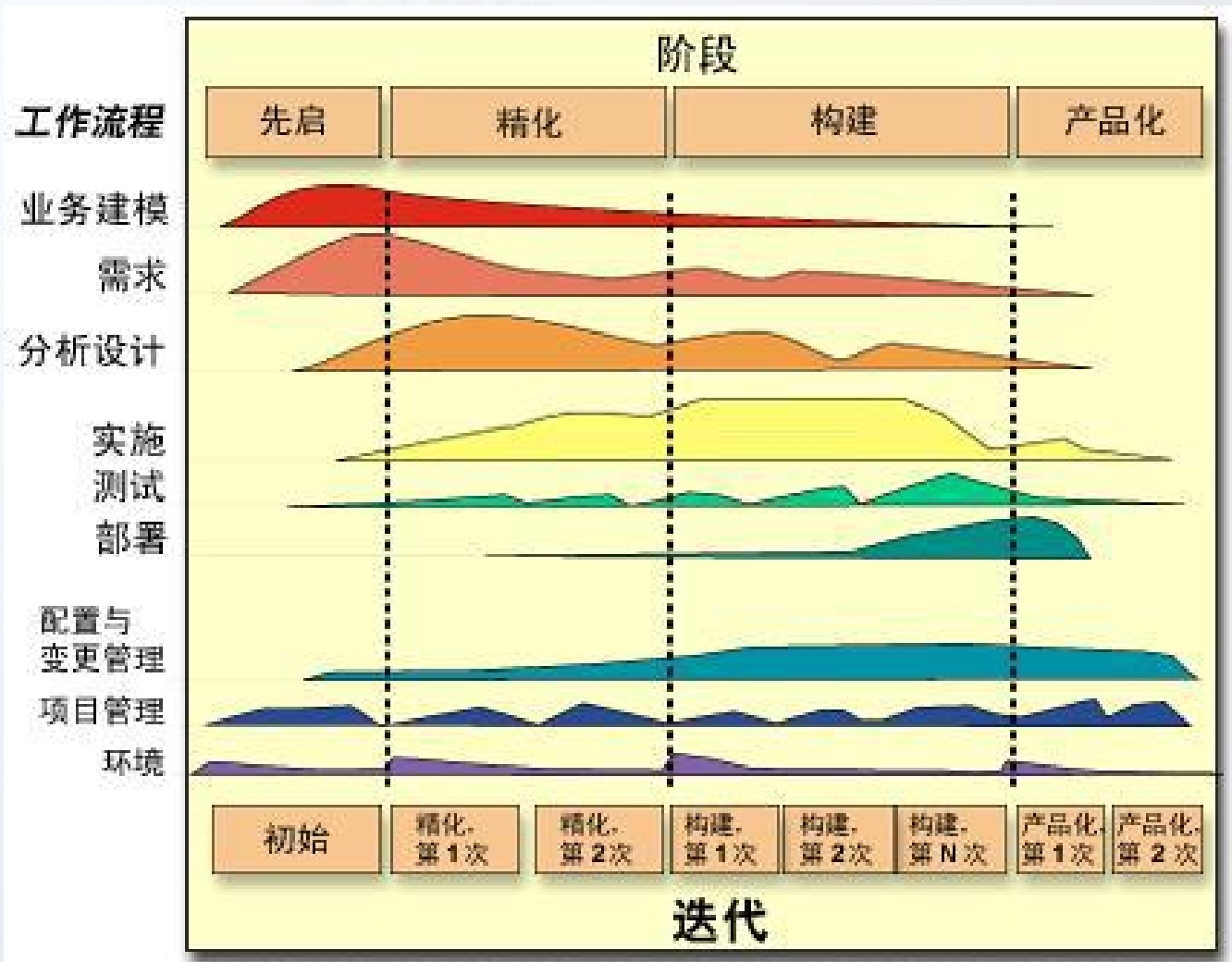
特点：

1. 自上而下、相互衔接的固定次序，如同瀑布流水，逐级下落。
2. 上一阶段的变换结果是下一阶段变换的输入，相邻两个阶段具有因果关系。

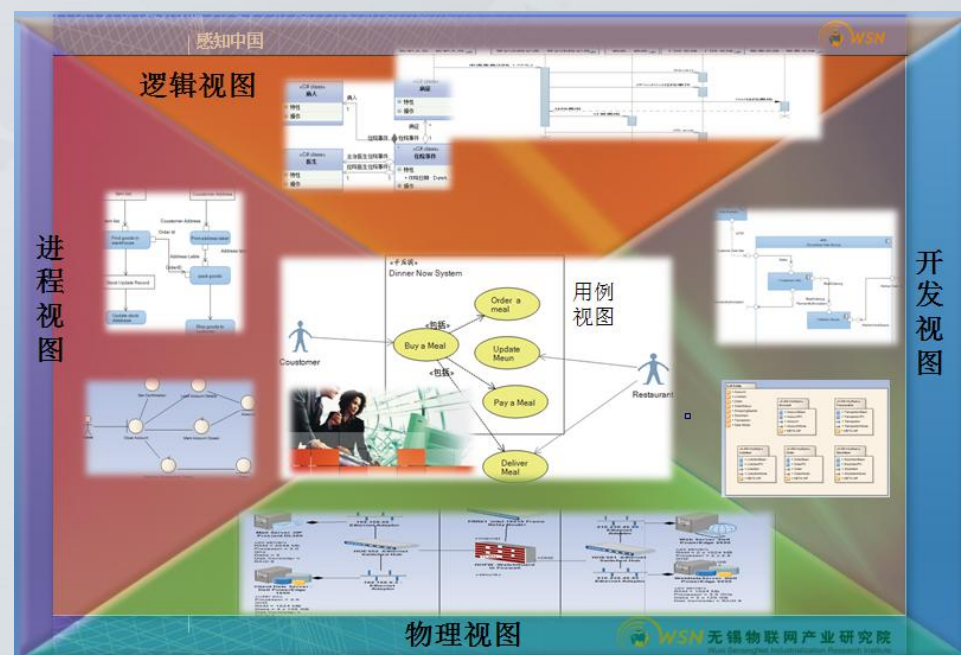
问题：

1. 各个阶段的划分完全固定，阶段之间产生大量的文档，极大地增加了工作量。
2. 开发模型是线性的，用户只有等到整个过程的末期才能见到开发成果，增加了风险。
3. 早期的错误可能要等到开发后期的测试阶段才能发现，进而带来严重的后果。

RUP统一软件过程 (Rational Unified Process) >>>

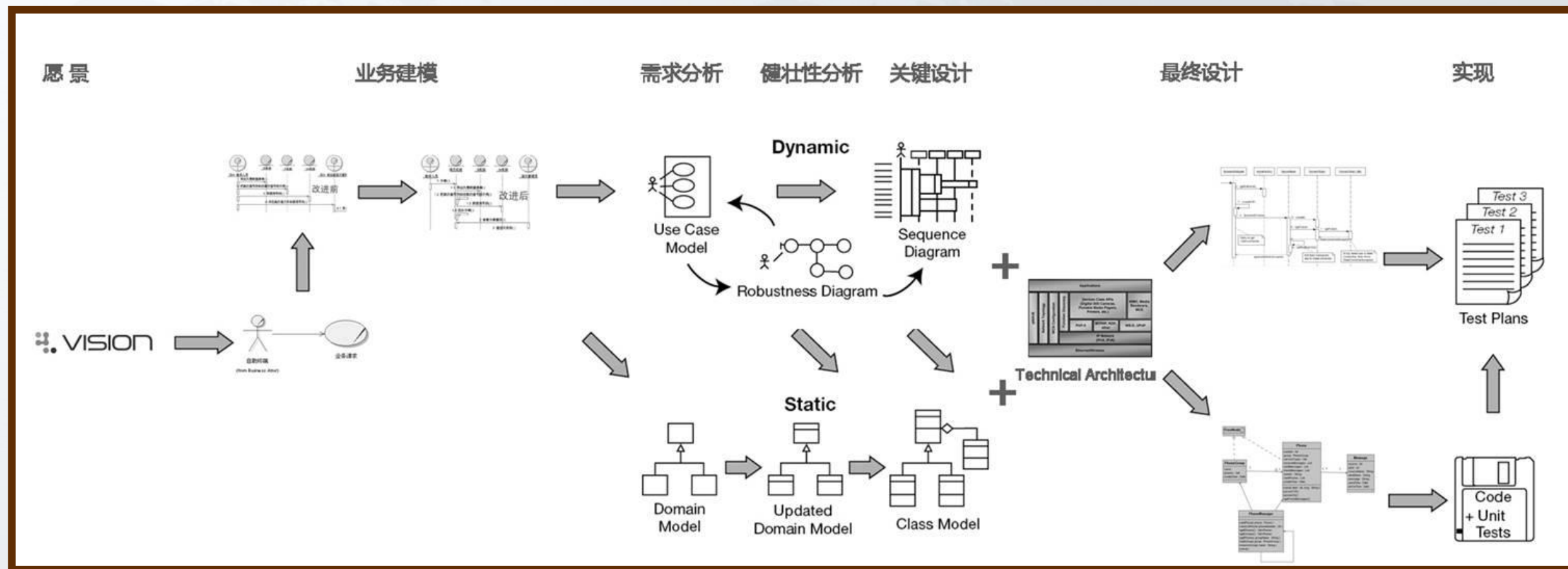


RUP的中心思想是：用例驱动、架构为中心、迭代和增量。



RUP 4+1视图

扩展ICONIX过程 >>>



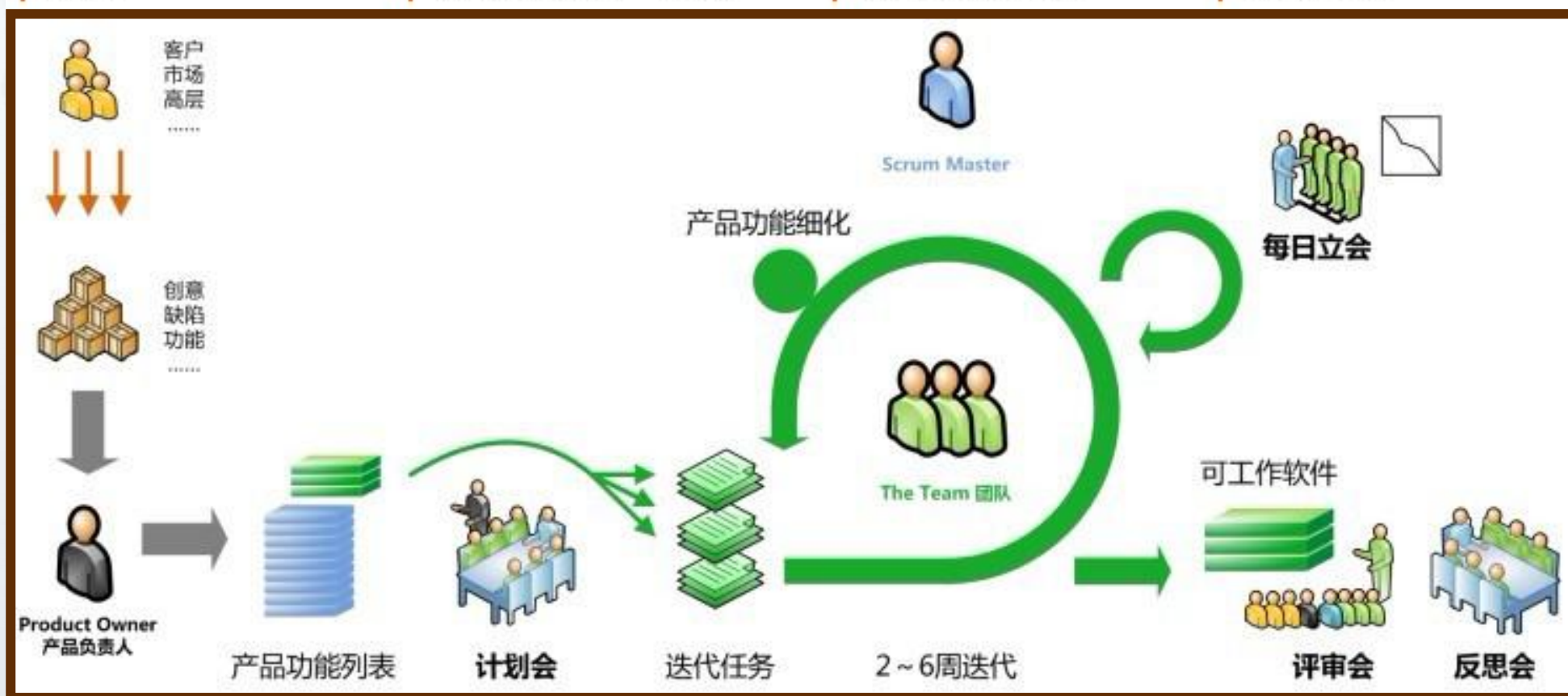
Scrum敏捷过程 >>>

产品负责人建立条目的产品待开发项，并进行优先级排序。

在迭代计划会上，产品负责人讲解本迭代要开发的条目，团队进行估算并放入下一个迭代。

团队在迭代内完成所列需求，每天都开每日“立”会以沟通进度和问题。

在迭代终点的迭代评审会上，团队向产品负责人等展示开发成果。



如何理解迭代与增量



增量：逐块建造

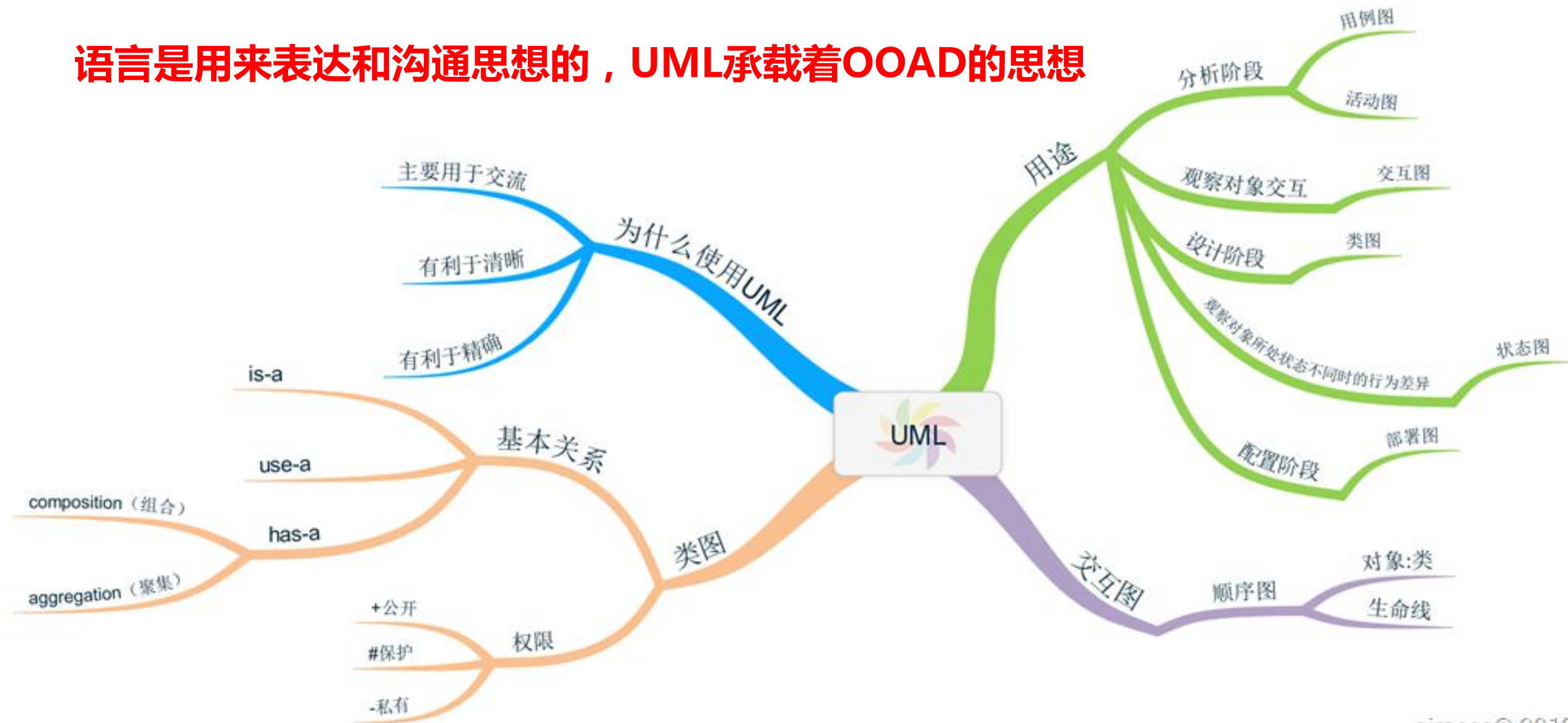


迭代：反复求精





语言是用来表达和沟通思想的，UML承载着OOAD的思想



统一建模语言（UML）



| | |
|--------|--|
| UML静态图 | 类图（Class Diagram）：模型化系统的结构 |
| | 对象图（Object Diagram）：对象及对象间的相互关系 |
| | 组件图（Component Diagram）：模型化组件的组织 and 依赖 |
| | 部署图（Deployment Diagram）：模型化系统的硬件分布 |
| UML动态图 | 时序图（Sequence Diagram）：模型化系统的行为 |
| | 协作图（Collaboration Diagram）：模型化系统的行为 |
| | 状态图（Statechart Diagram）：模型化状态相关的方面 |
| | 活动图（Activity Diagram）：模型化系统内的事件流 |
| | 用例图（Use Case Diagram）：模型化系统与外界的交互 |

目录 >>>

一

软件工程学科概览

二

IT行业人才格局及成长路线

三

软件过程—软件工程的核心组成部分

四

本课程的要点说明

课程价值 >>>

1

团队合作

如何从单兵作战转为团队合作，完成团队的协调与管理.....

2

软件过程

掌握软件从无到有的整个过程.....
Scrum、ICONIX

3

过程方法

实用的软件过程推演方法
用户故事、故事点、燃尽图、用例.....

4

实用工具

实用的工具软件
Axure、EA.....

课程考核 >>>



必修课，4学分。

学习建议 >>>

理解思想

掌握过程

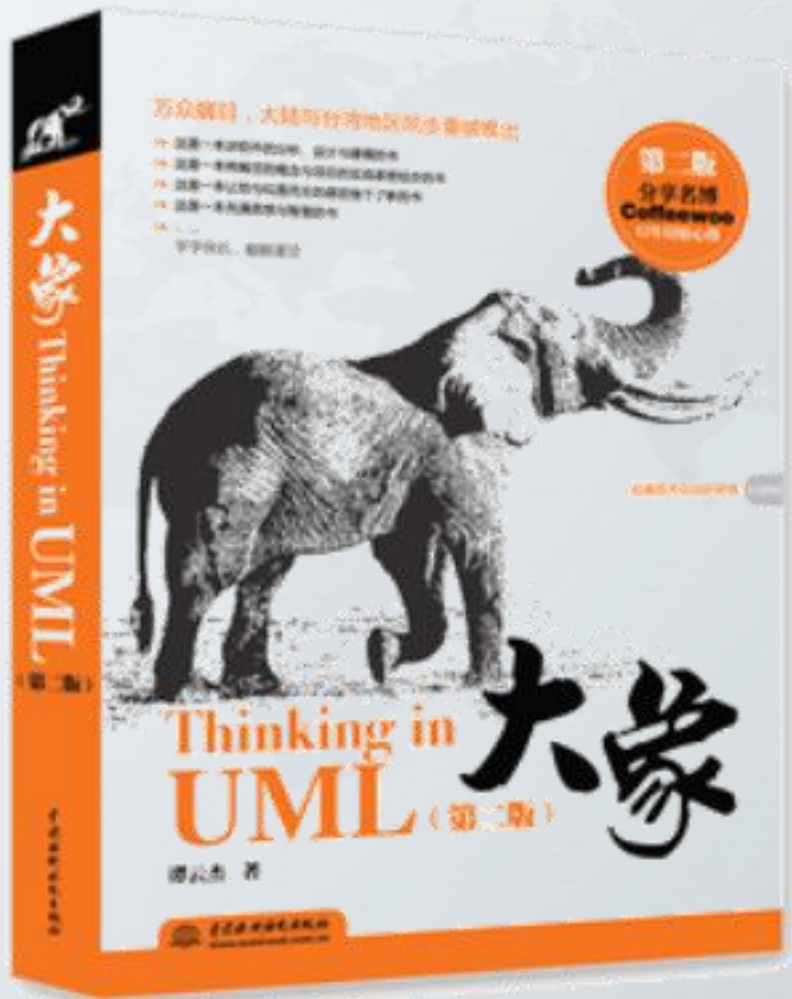
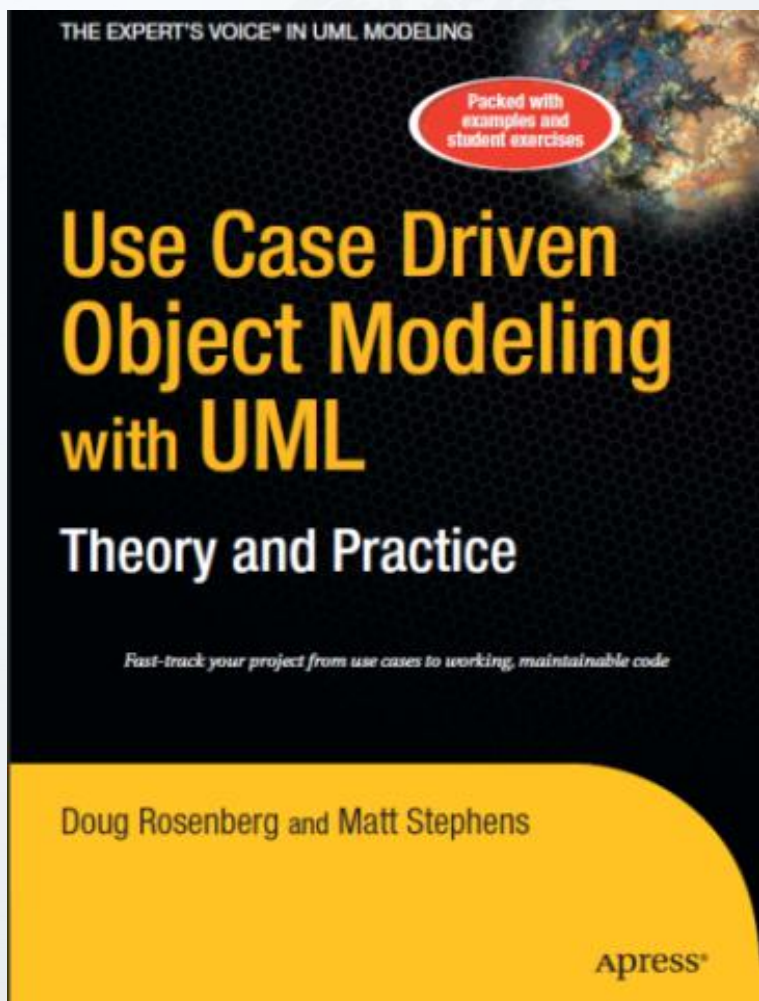
掌握方法

了解工具

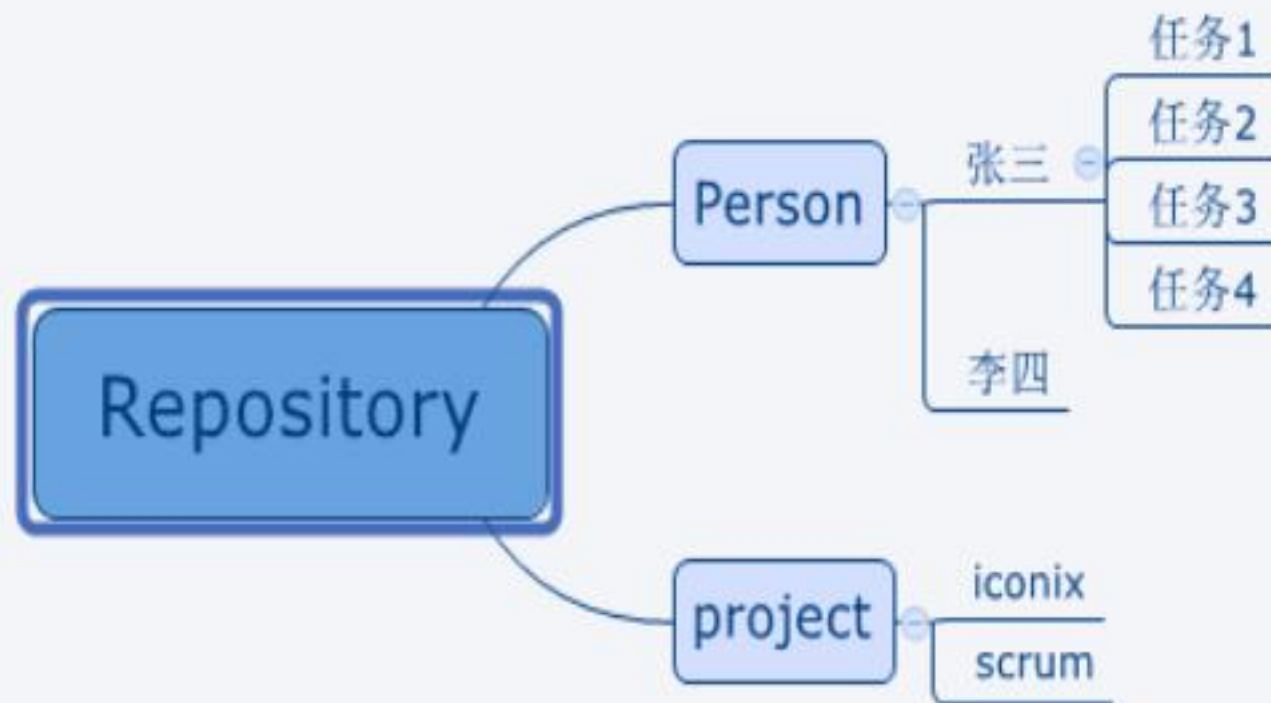
思想是重点；过程是方式；方法和工具是载体。

练中学、错中学，反复实践

参考书目 >>>



如何提交作业 >>>



1. 使用GitHub进行作业管理
2. 以组为单位，每组建立一个代码仓库（组长提供链接地址）
3. 仓库中创建若干文件夹，文件夹名与组员姓名一一对应，用来提交各自的作业
4. 每个组员完成作业后将其提交到仓库中对应自己的文件夹中

总结：匠人 vs. 大师

- 视野、文化、创新





THANKS