

# 第一章 软件工程概述

## 目录 >>>







## 目录 >>>







### 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"?



### 演员是不是电影的全部?







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%

•技术的影响因素有多大?

•非技术的因素靠什么决?



### 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)

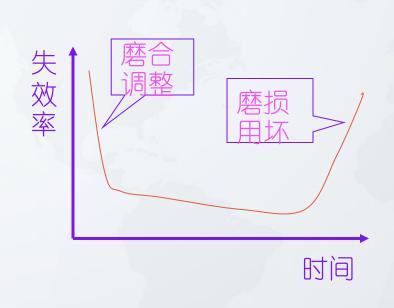
### 软件特点 >>>

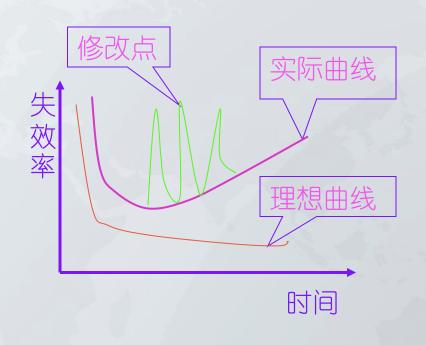
### 2、软件特点

- 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。

## 软件特点







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) 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) 软件文档资料不完整、不合格。
  - (5) 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。
  - (6) 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。
- (7) 软件产品的特殊性和人智力的局限性,导致人们无力处理"复杂问题"。 所谓"复杂问题"的概念是相对的,一旦人们采用先进的组织形式、开发方法 和工具提高了软件的开发效率和能力,新的、更复杂的问题又摆在人们面前。

### 软件危机的原因:

- (1) 用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,这种差异必然导致开发出来的软件产品与用户要求不一致。
- (3) 软件开发人员不能有效地、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。
- (4) 大型软件项目需要组织一定的人力共同完成,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。
- (5) 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。

### 软件工程的概念及范畴



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



#### 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<mark>用户需求</mark>,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

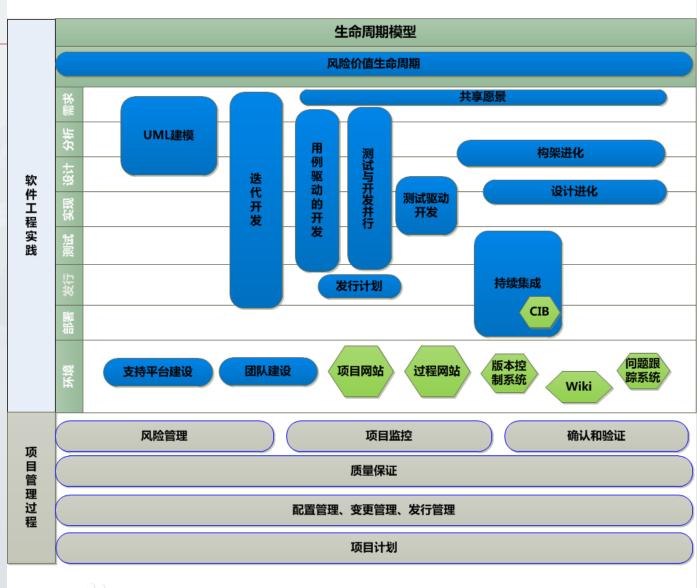
- (8) 软件工程过程
- (9) 软件工程工具和方法
- (10) 软件质量

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"

#### 内容概览



图例:

工具

关键过程

最佳实践

## 目录 >>>







## 学成文武艺, 货与帝王家 >>>

• IT行业的从业出路是什么?



上午会议结束。

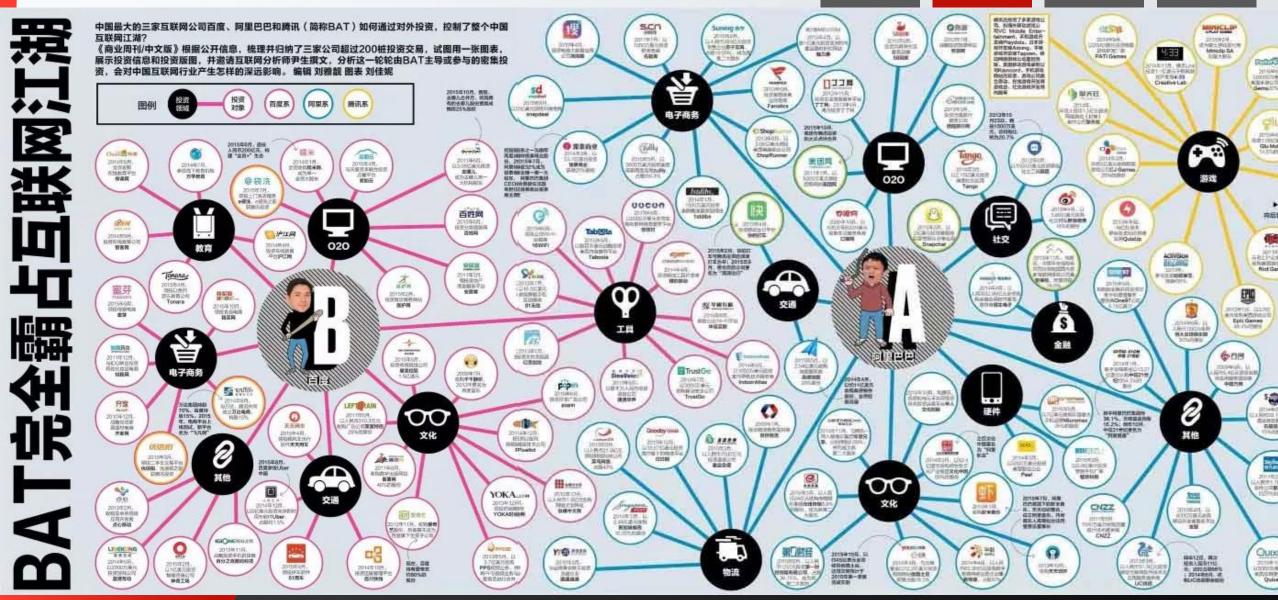
马云:"去帮我买肯德基和必胜客吧,一会儿还要开会"

秘书: "马总,已经买好了, 共计4.6亿,您签下付款



### 看看今天的IT行业版图





### BAT的企业基因? >>>

商业线

你会选择谁?

功用性 产品线 [capability] 称许性 可行性 [desirability] [viability]

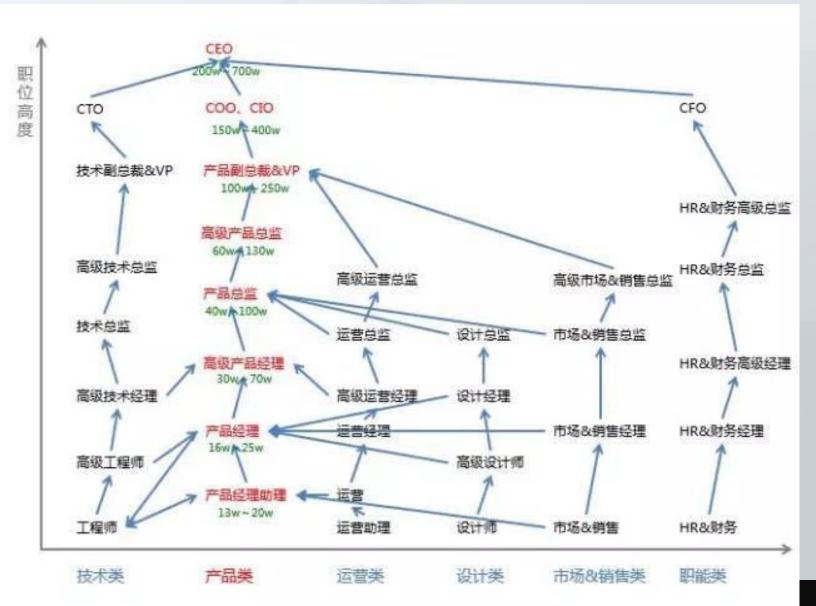
Doblin Group 公司总裁Larry Keely

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







































































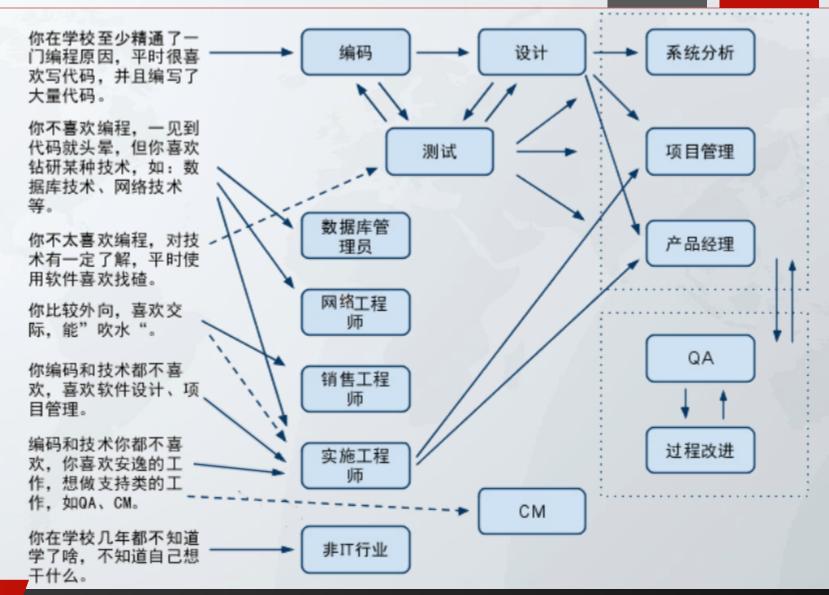




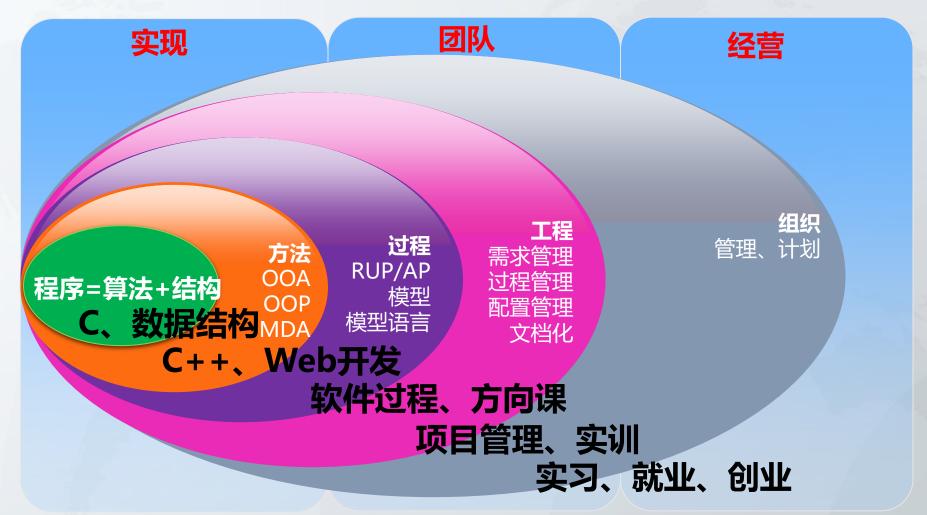
### 传统IT公司的人才格局 >>>



### 传统IT人才成长路径 >>>



### 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》

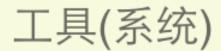
## 目录 >>>







## 软件工程三要素 >>>



在分析业务过程中发 现软件的价值

MoinMoin XMind JIRA

Visual Paradigm/StarUML Axure RP 使开发人员和用户就 系统做什么"达成共识

### 方法(技能)

业务建模方法[组织用例图/业务序列图] 需求方法[系统用例识别和书写规约] 项目管理方法 配置管理方法

用来保证系统的 功能和性能

### 开发过程(框架)

Iterative and incremental/development

 $_{
m UML}$ 

Unified Processe/Agile Processes/...

根据客户、团队和项目特征订制框架 关键是对核心活动的选取和定义,如业务建模/需求/分析/设计/实施/发布

## 瀑布模型 (Waterfall Model) >>>





#### 特点:

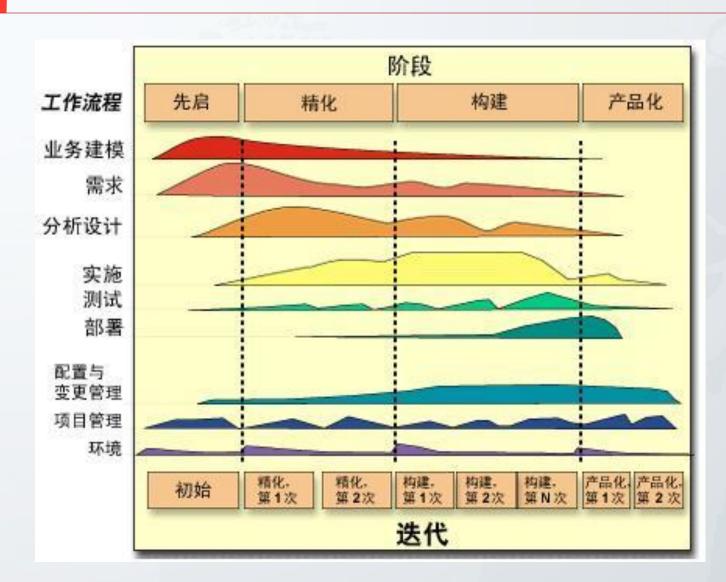
- 1. 自上而下、相互衔接的固定次序,如同 瀑布流水,逐级下落。
- 2. 上一阶段的变换结果是下一阶段变换的 输入,相邻两个阶段具有因果关系。

#### 问题:

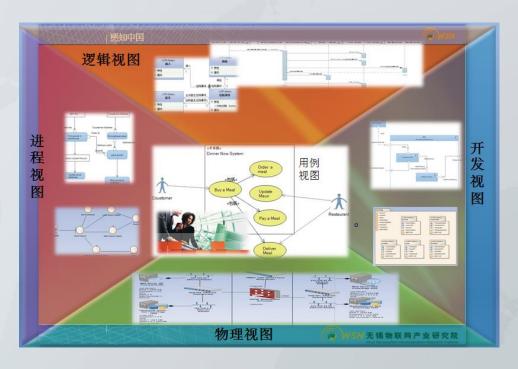
- 1. 各个阶段的划分完全固定, 阶段之间产生 大量的文档,极大地增加了工作量。
- 2. 开发模型是线性的,用户只有等到整个过 程的末期才能见到开发成果,增加了风险。
- 3. 早期的错误可能要等到开发后期的测试 阶段才能发现,进而带来严重的后果。

## 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。

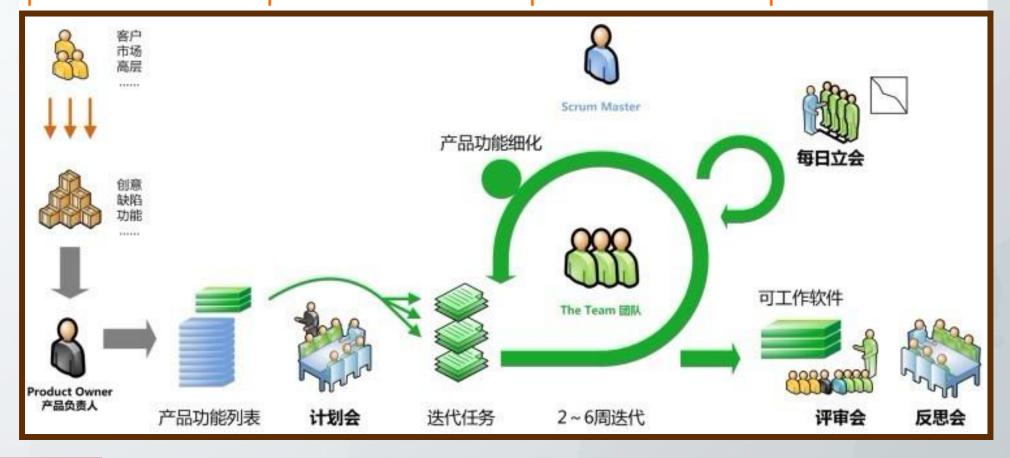


RUP 4+1视图

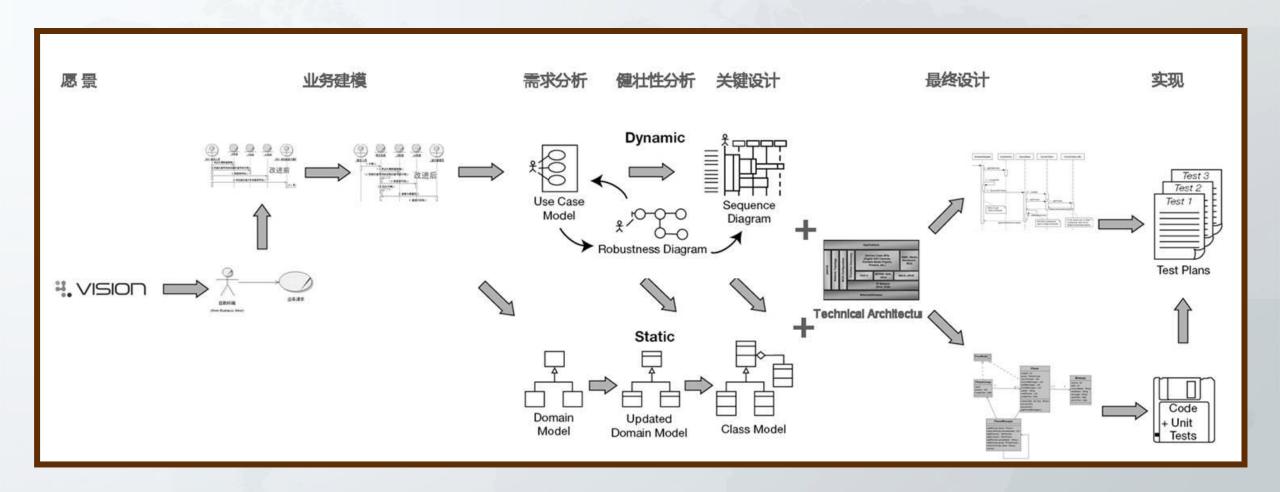
#### Scrum敏捷过程 >>>

产品负责人**建立条目化的产 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。



### 扩展ICONIX过程 >>>

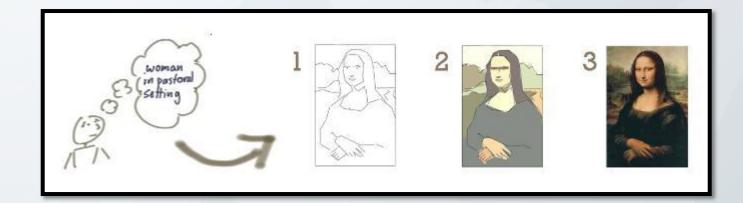


## 如何理解迭代与增量 >>>



增量:逐块建造

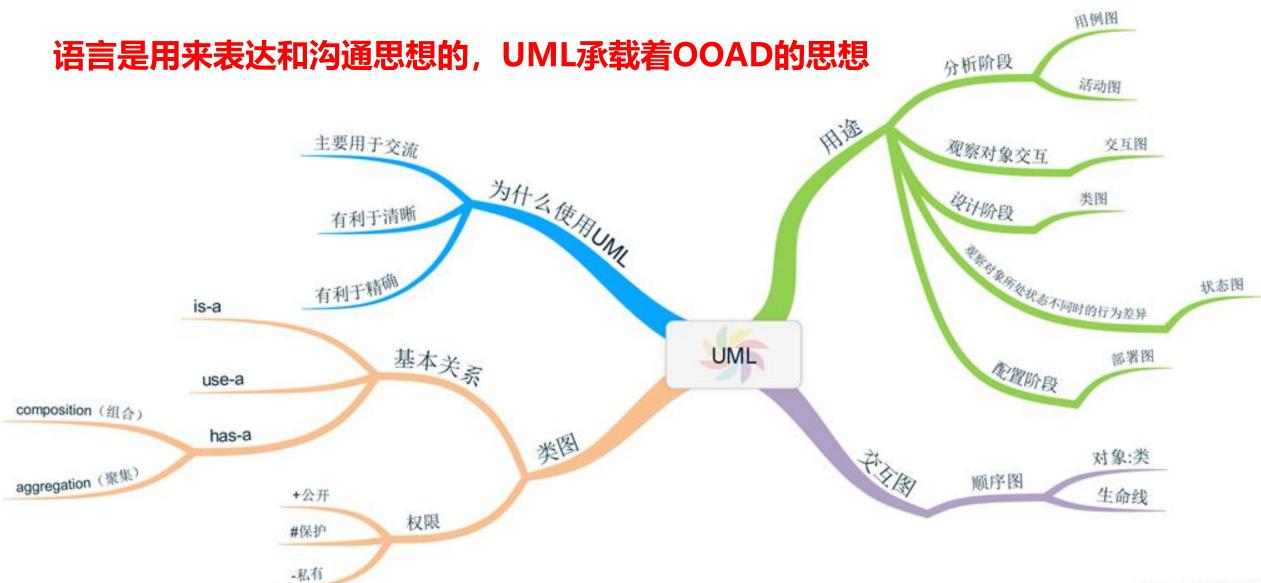




迭代: 反复求精

#### 统一建模语言 (UML, UnifiedModelingLanguage)





# 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):模型化组件的组织和依赖
	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 >>>





### 课程价值 >>>

如何从单兵作战转为团队合作,完成 团队的协调与管理.....

掌握软件从无到有的整个过程..... 软件过程 Scrum、ICONIX

实用的软件过程推演方法 用户故事、故事点、燃尽图、用例.....

实用的工具软件 实用工具 Axure、EA......

# 课程考核 >>>



必修课, 3学分。

缺勤或违反课堂纪律一次扣2分

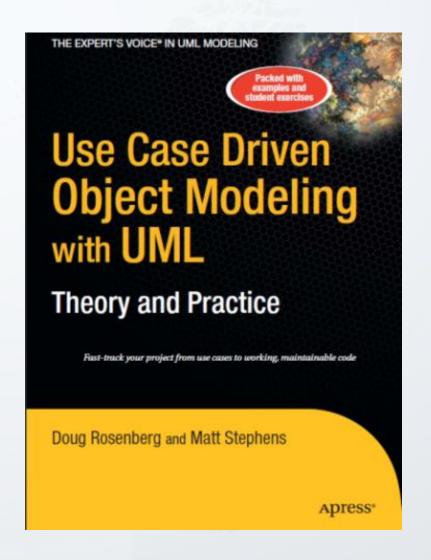
### 学习建议 >>>

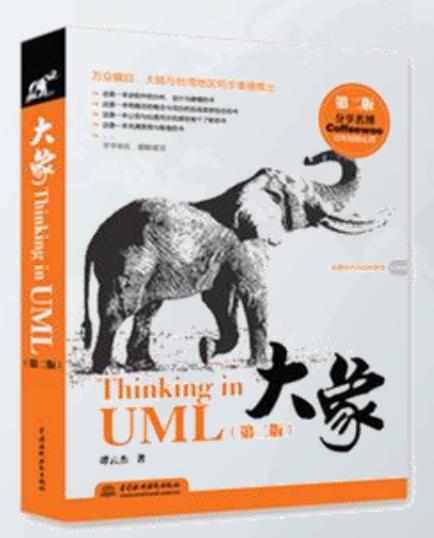


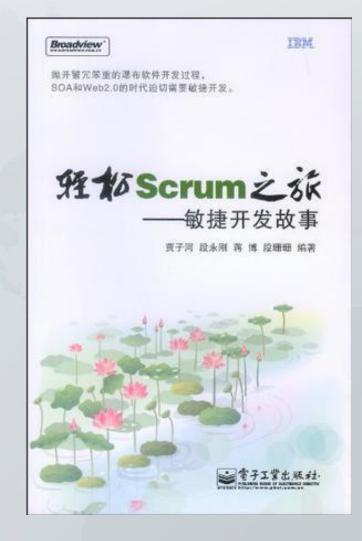
思想是重点; 过程是方式; 方法和工具是载体。

练中学、错中学, 反复实践

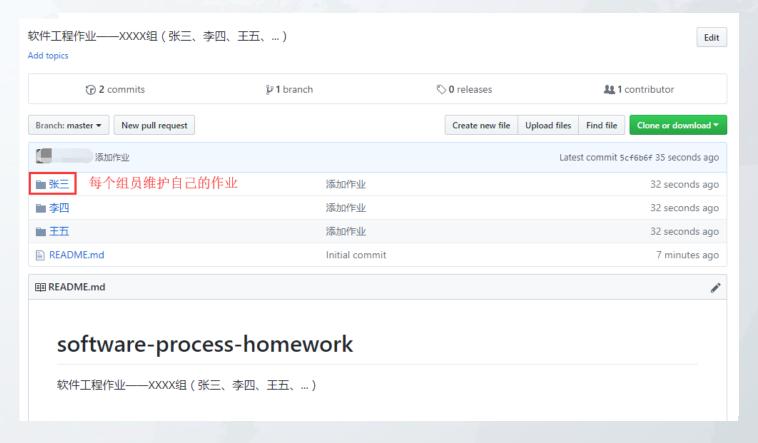
#### 参考书目 >>>







#### 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——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





