



软件开发环境国家重点实验室
State Key Laboratory of Software Development Environment

软件团队和流程

团队 vs. 工作组

- **工作组**

- 为了完成某个任务临时聚拢在一起的一群人
- M个人，各自把N块砖从A地运到B地

- **团队**

- 有一致的目标，团队要一起完成目标
- 团队成员有各自分工，互相依赖合作，共同完成任务
- M个人，负责共同把 $N \times M$ 块砖从A地运到B地

- **有的事情只有团队才能完成，工作组不可能做到**

- **团队能够容忍需求的变更和困难，而工作组通常做不到**

团队？工作组？

- **一个工作组的成员可以被各个击破**

- ❑ 当初他们屠杀工会人士，我没有说话，因为我不是工会人士
- ❑ 后来他们屠杀共产党，我也没有说话，因为我不是共产党
- ❑ 后来他们杀犹太人，我还是没有说话，因为我不是犹太人
- ❑ 再接下来，他们杀天主教徒，我仍然保持沉默，因为我是基督教徒
- ❑ 最后他们要杀我了，已经没有人为我说话了，因为能够说话的人都被他们杀光了





问题：如何组建一支高效的团队？采用什么管理模式？



[team]

一个人的能力再强也
无法顾及所有的事情，
必须得有一个很强的团队



团队

• 主治医师模式 (Chief Programmer Team)

- 首席程序员负责主要模块的设计和编码，其他成员从各种角度支持他的工作 (后备程序员、系统管理员、工具开发、编程语言专家、业务专家)
- 例子：
 - Frederic Brooks Jr., IBM System 360的开发

• 明星模式 (Super-star Model)

- 主治医师模式运用到极点，就蜕变为明星模式
- 例子：
 - 刘翔——翔之队

• 社区模式 (Community Model)

- 由很多志愿者参与，每个人参与自己感兴趣的项目，大部分人不拿报酬
- “众人拾柴火焰高”，只索取不回报将会导致项目的失败
- 社区不意味着随意，成功的项目都有很严格的代码复审和质量控制机制
- 例子：
 - Linux、Apache

• 业余剧团模式 (Amateur Theater Team)

- 在每一个项目中，不同的人会挑选不同的角色，下一次可能会换一个完全不同的角色，听从一个中央指挥的指导和安排
- 例子：
 - 学术实践项目或培训项目

• 秘密团队 (Skunk works Team)

- ❑ 团队在秘密状态下进行开发，别人不知道他们具体在做什么
- ❑ 团队内部有极大的自由，没有外界的干扰，团队成员有极大的投入
- ❑ 例子：
 - Apple Macintosh系统

• 特工团队 (SWAT Team)

- ❑ 有一些有特殊技能的专业人士组成
- ❑ 解决一些棘手而又紧迫的问题
- ❑ 例子：
 - Y2K问题、网站安全性服务团队

• 交响乐团模式 (Orchestra Team)

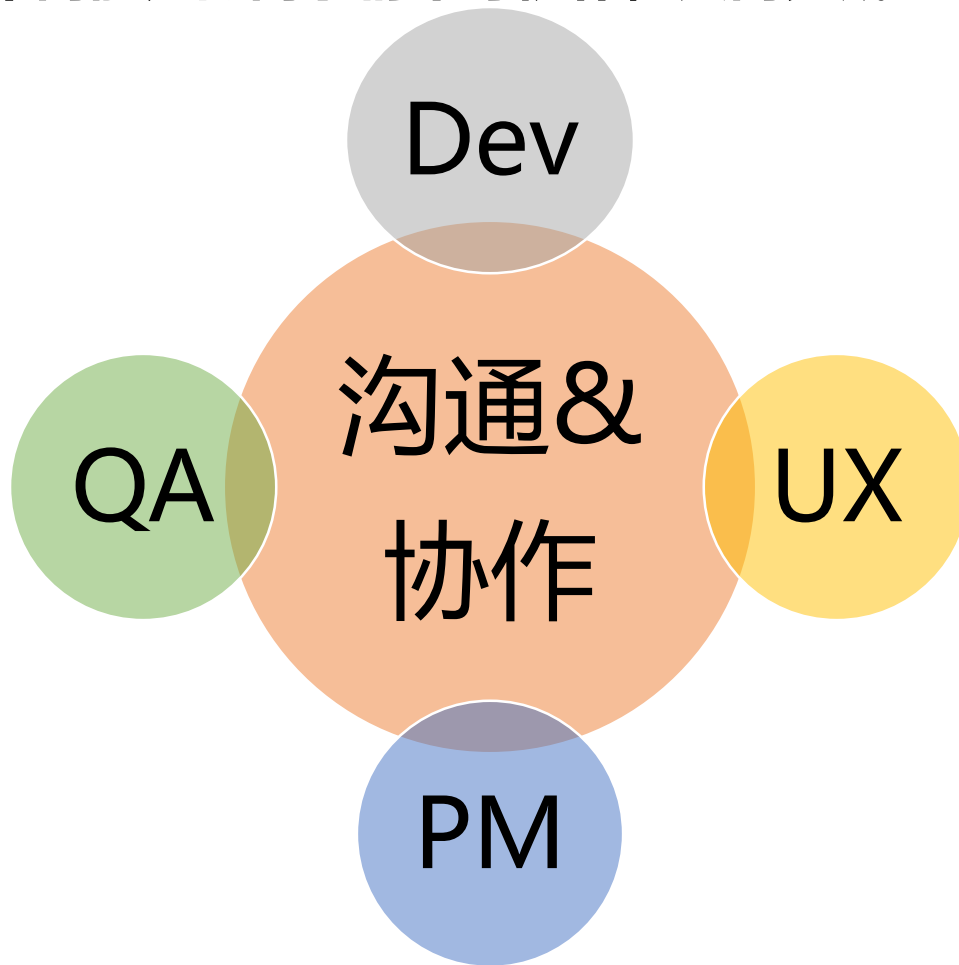
- ❑ 家伙多，门类齐全
- ❑ 各施其职，有专门的场地
- ❑ 演奏都靠谱，同时看指挥
- ❑ 演奏的曲目经过多次练习，重在执行
- ❑ 例子：
 - 大型软件公司的开发团队，比如Office软件

• 爵士乐模式 (Jazz Band)

- ❑ 不靠谱，没有现场指挥
- ❑ 强调个性化的表达，强有力的互动，对变化的内容有创意的回应

• 功能团队模式 (Feature Team)

- 具备不同能力的同事们平等协作，共同完成一个功能

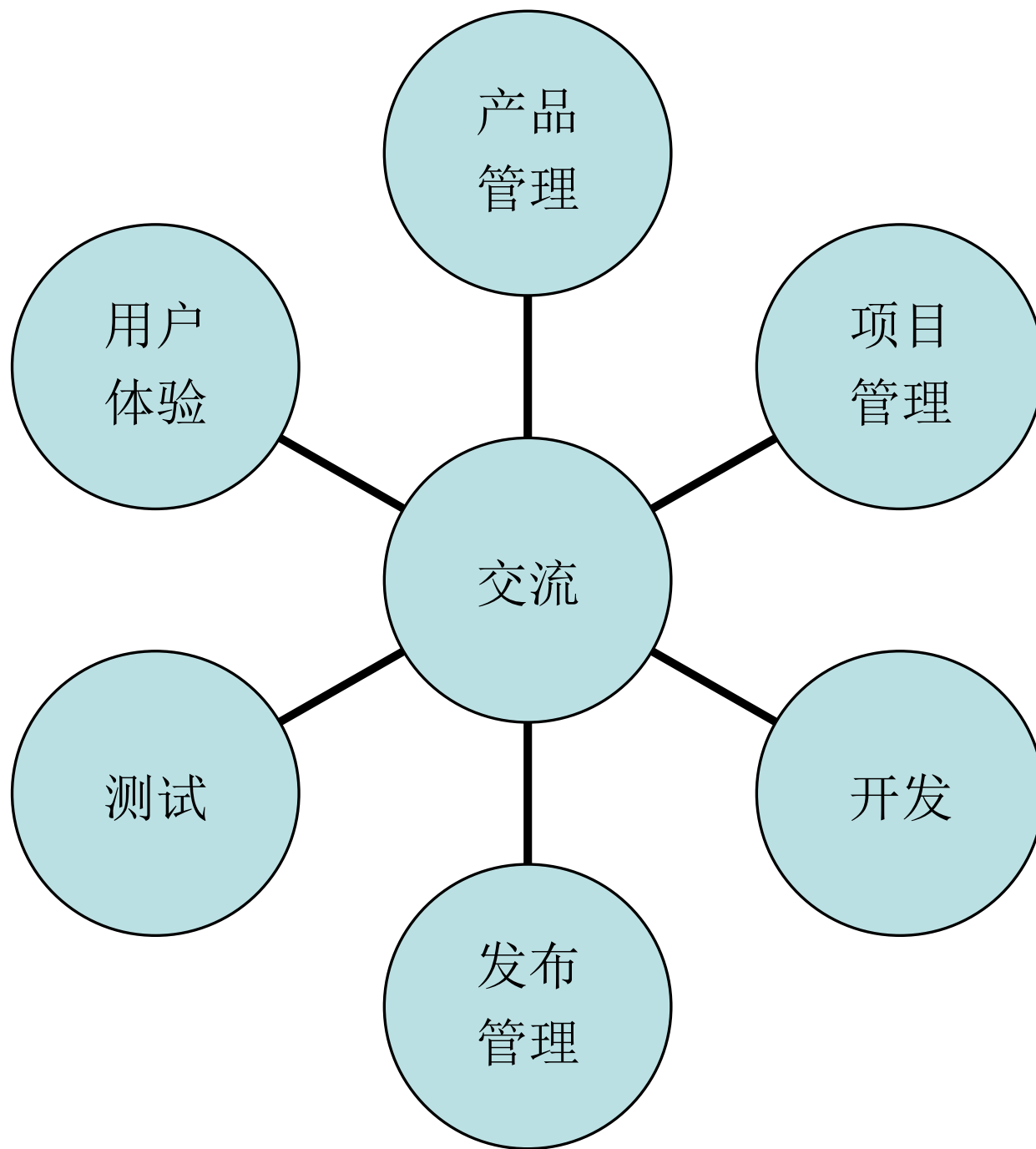


• 官僚模式 (Bureaucratic Model)

- 类似于大机构的组织架构，层层报告
- 成员之间不仅有技术上的合作和领导关系，还有组织上的领导关系
- 跨组织合作比较困难



你们选择什么样的团队结构？



- 写了再改模型(Code-and-fix)
- 瀑布模型(Waterfall Model)
- 瀑布模型的各种变形
- 快速原型模型
- 螺旋模型 (Spiral Model)
- Rational统一流程 (Rational Unified Process)
- 老板驱动的流程 (Boss-Driven Process)
- 渐进交付模型 (Evolutionary Delivery)
- 敏捷开发



写了再改模型



- 常用的模型、简单，但是不一定好用

• 优点

- 没有额外开销
- 不需要太多准备或相关知识

• 缺点

- 不能对过程进行评估
- 无法对质量进行评估

• 适合于

- 小规模的项目
- 概念验证程序
- 不实用的演示程序
- 看过就扔的原型

瀑布模型



软件概念

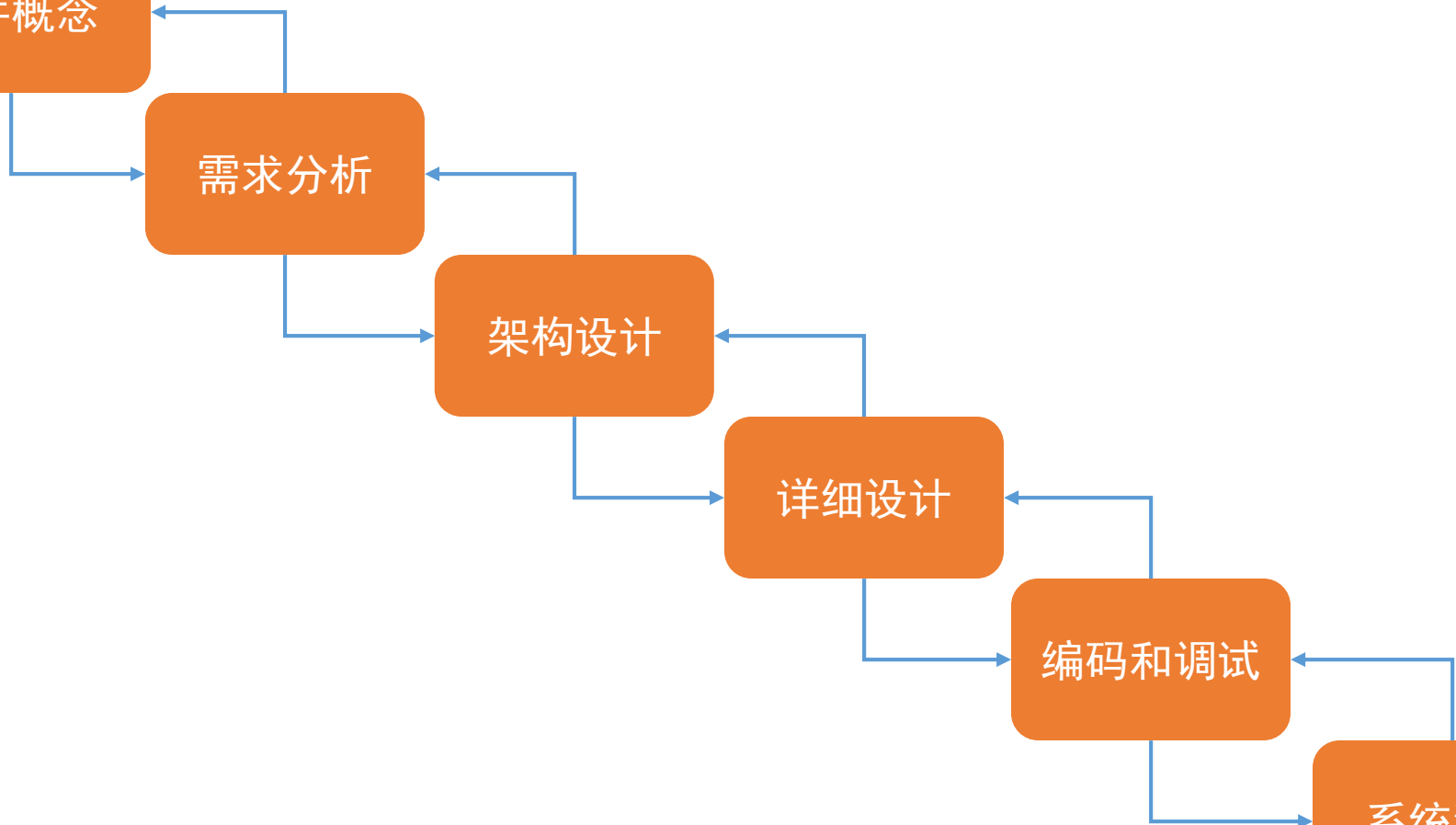
需求分析

架构设计

详细设计

编码和调试

系统测试



瀑布模型

- 它反映了人类解决问题的思路

- 优点

- 每一步的结果可验证
- 降低风险
- 提供结构

- 缺点

- 各步骤之间完全分离，与软件生成实际不符
- 回溯修改很困难甚至不可能，不适应软件生成中时时回溯的需求
- 最终产品知道最后才出现

- 适合于

- 产品的定义非常稳定，但产品的正确性非常重要 (Apollo Project)
- 产品模块之间的接口、输入和输出能很好的用形式化方法定义和验证
- 使用的技术非常成熟
- 负责各个步骤的子团队分属不同机构，不能频繁交流

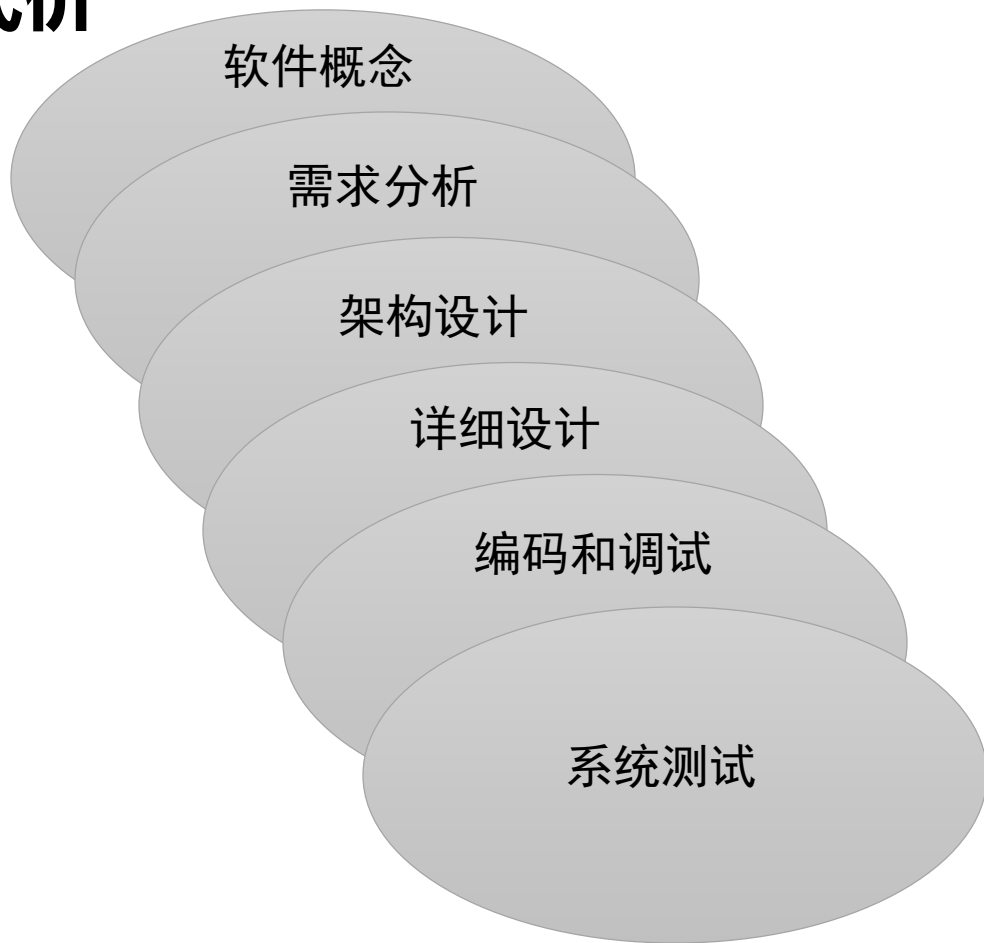
例子：制造汽车

- 你(用户) 提出要发动机、车身、车窗、方向盘、加速踏板、刹车、手刹、座位、车灯
- 生产商按照瀑布模型流程给你生产，六个月后交付
- 看到样车后...
- 你提出 – 我当初忘了一件小事，要有倒车灯
 - 当倒车的时候，倒车灯会亮
- 生产商说：
 - 我要重新设计车尾部，加上倒车灯，把车底拆开，安装线路，修改传动装置把倒车档和倒车灯联系起来……我得重新开始
- 你说：这不是很小的一件事么？

- 生鱼片 (Sashimi)



- 步骤之间有重叠
- 降低了步骤完全分离的代价
- 缺点
 - ❑ 里程碑重叠
 - ❑ 并行性可能导致沟通错误

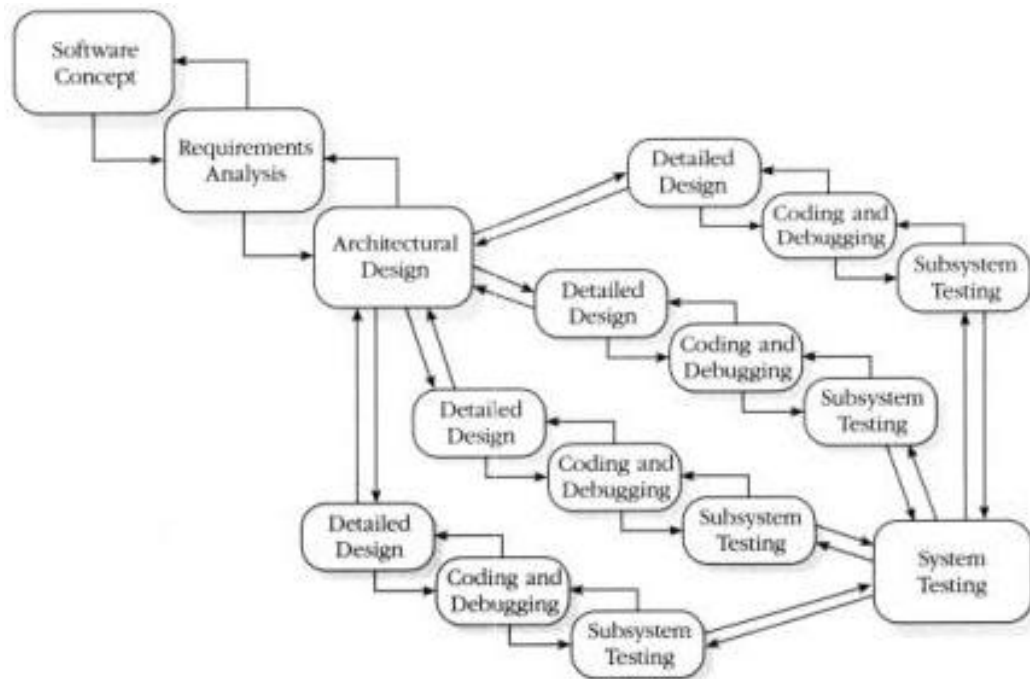


子瀑布模型

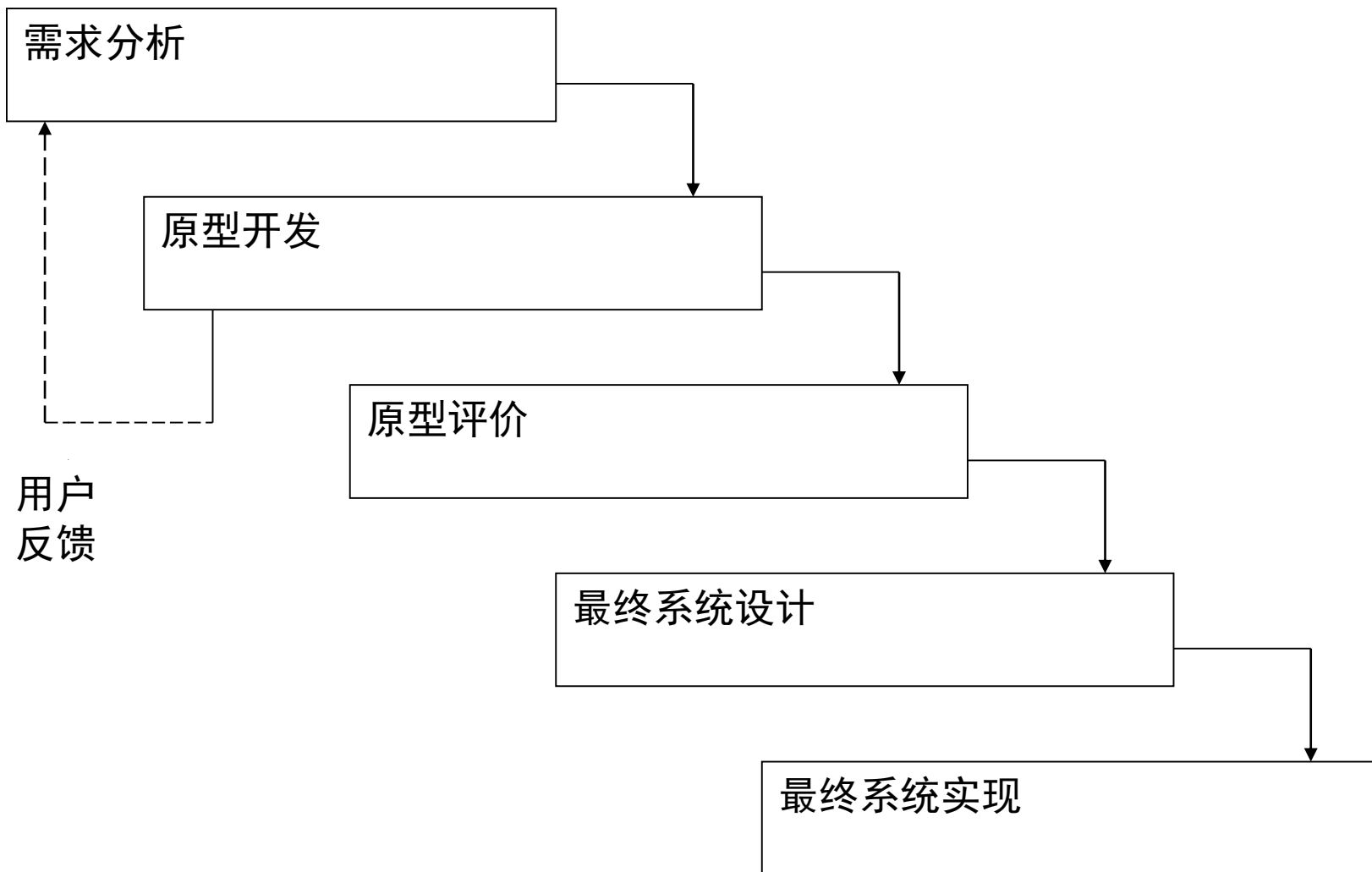
- 分治法

- 风险

- 不可预见的互依赖性
- 集成的代价大



快速原型模型



• 特点

- ❑ 快速开发工具，与用户有好的交互性
- ❑ 循环：修改-评价-再修改
- ❑ 低成本

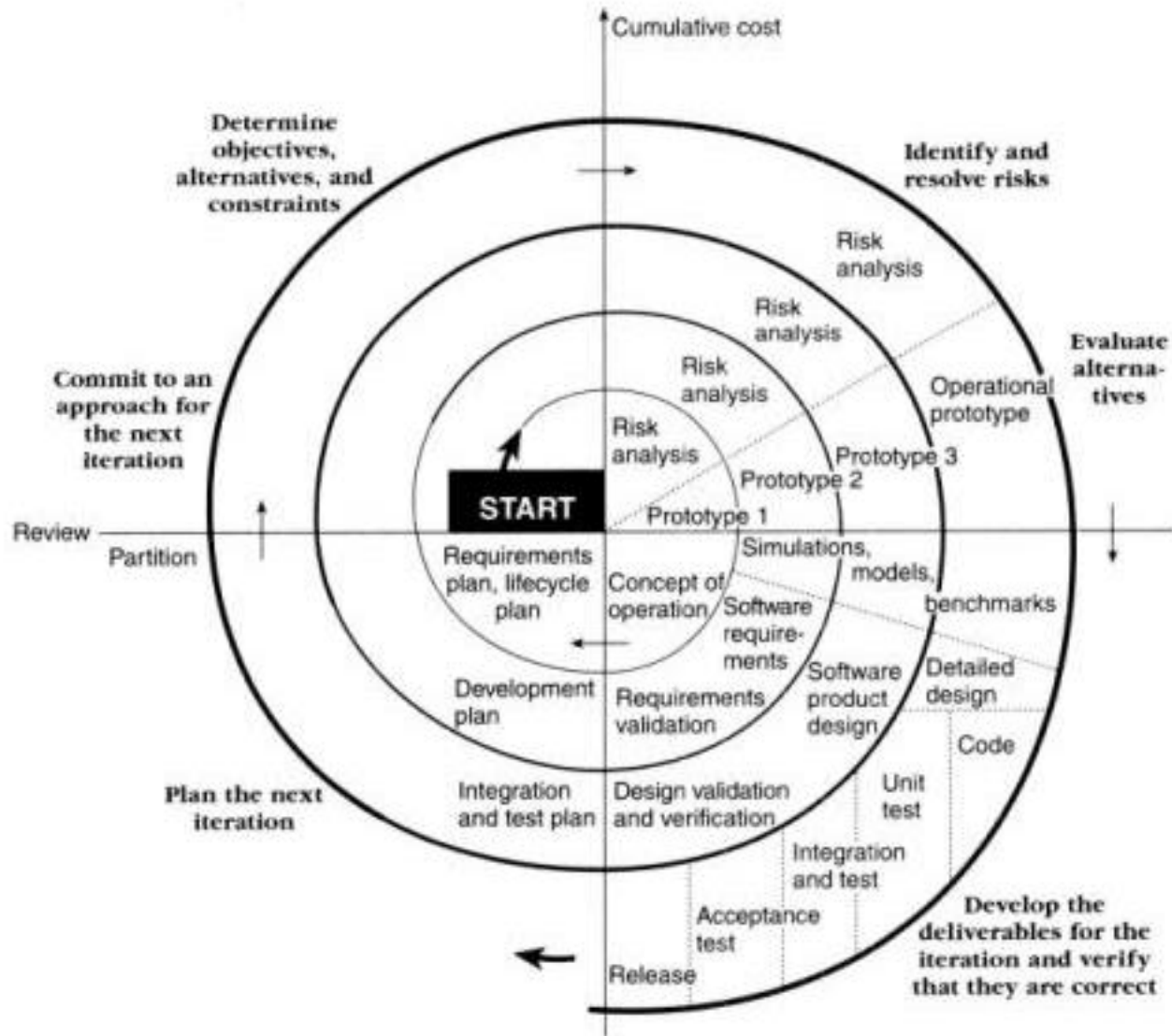
• 种类

- ❑ 渐进型
- ❑ 抛弃型

• 缺点

- ❑ 不适合大规模系统的开发，开发过程管理要求高，整个开发过程要经过“修改—评价—再修改”的多次反复；
- ❑ 用户过早看到系统原型，误认为系统就是这个模样，易使用户失去信心，开发人员易将原型取代系统分析；
- ❑ 缺乏规范化的文档资料

螺旋模型



• 优点

- 能够在每一轮控制风险
- 随着消耗的增加，风险降低

• 缺点

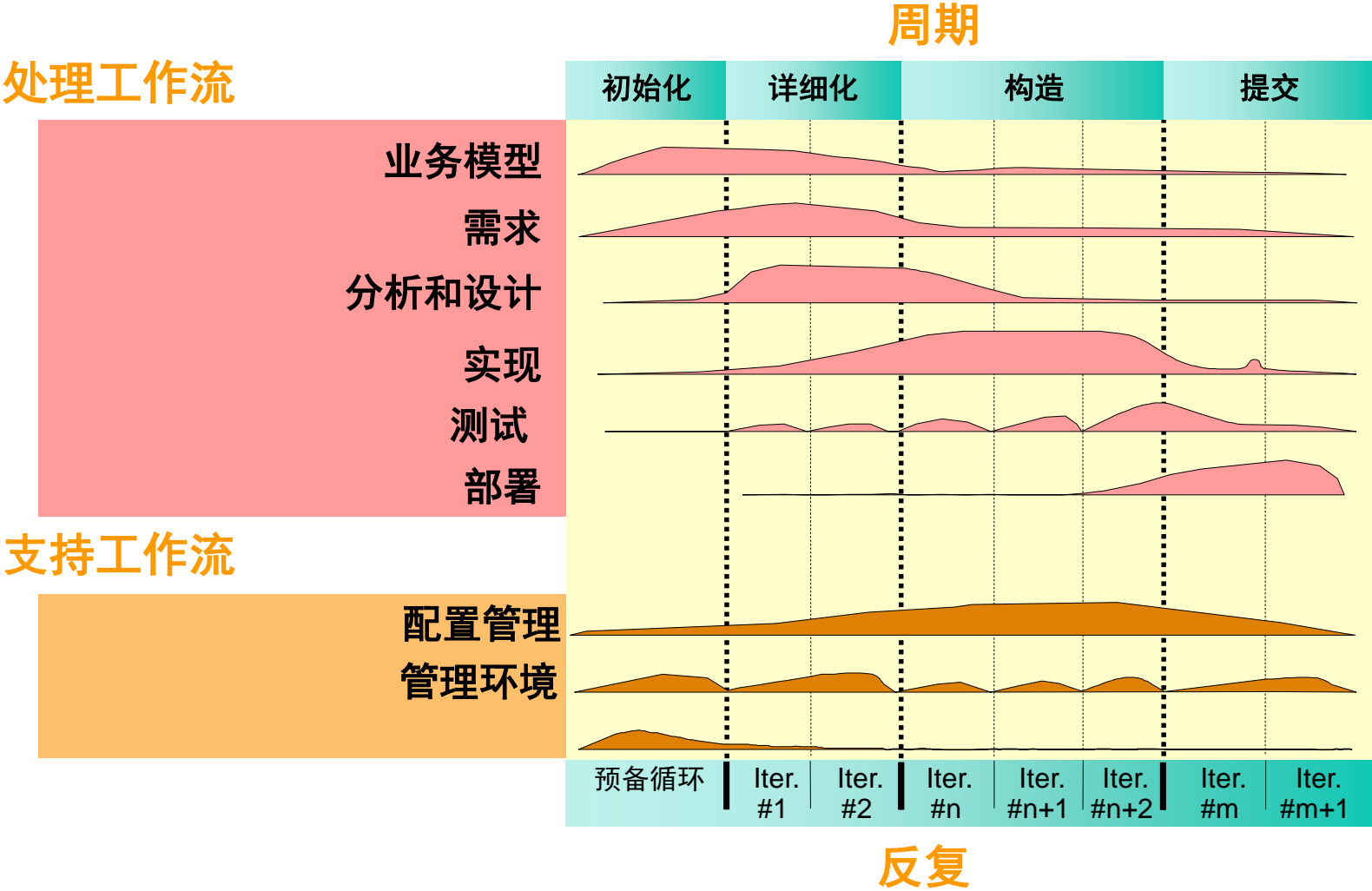
- 复杂，需要小心谨慎、细致和知识广博的管理
- 很难定义目标和可验证的里程碑

• 适合于

- 具有很多不确定性的复杂项目，并且具有知识广博、技能丰富的团队

- Rational统一过程 workflow
 - 业务建模
 - 需求
 - 分析和设计
 - 实现
 - 测试
 - 部署
 - 配置和变更管理
 - 项目管理
 - 环境
- Rational统一过程的四个阶段
 - 初始阶段
 - 细化阶段
 - 构造阶段
 - Beta版本
 - 交付阶段
 - 迭代到release版本

RUP的统一过程结构



统一软件过程RUP模型

- **RUP的时间轴被分解为四个顺序的阶段，分别是：**
 - 初始阶段(Inception)、细化阶段(Elaboration)、构造阶段(Construction)、交付阶段(Transition)
 - 每个阶段结束于一个主要的里程碑(Major Milestones)
 - 每个阶段本质上是两个里程碑之间的时间跨度
 - 在每个阶段的结尾执行一次评估以确定这个阶段的目标是否已经满足。如果评估结果令人满意的话，可以允许项目进入下一个阶段。
- **RUP的阶段目标**
 - **初始阶段**的目标是为系统建立业务案例并确定**项目的边界**。
 - **细化阶段**的目标是分析问题领域，**建立健全的体系结构基础**，编制项目计划，淘汰项目中最高风险的元素。
 - **构建阶段**，所有剩余的构件和应用程序功能被**开发并集成**为产品，所有的功能被详细**测试**。
 - **交付阶段**的重点是确保软件对最终用户是可用的。

• RUP模型的优点

- 提高了团队生产力，在迭代的开发过程、需求管理、基于组件的体系结构、可视化软件建模、验证软件质量及控制软件变更等方面，针对所有关键的开发活动为每个开发成员提供了必要的准则、模板和工具指导，并确保全体成员共享相同的知识基础
- 建立了简洁和清晰的过程结构，为开发过程提供较大的通用性。

• RUP模型的缺点

- RUP在理论上，是比较理想的，但在实际应用上，还需要更多的工具的支持和普及推广工作。

• 软件开发流程由行政领导主导

- ❑ 软件订单的获得不是主要靠技术实力
- ❑ 大型企业内部，软件功能往往由行政体系来决定
- ❑ 有些老板比一般技术人员更懂市场和竞争
- ❑ 软件团队尚未成熟

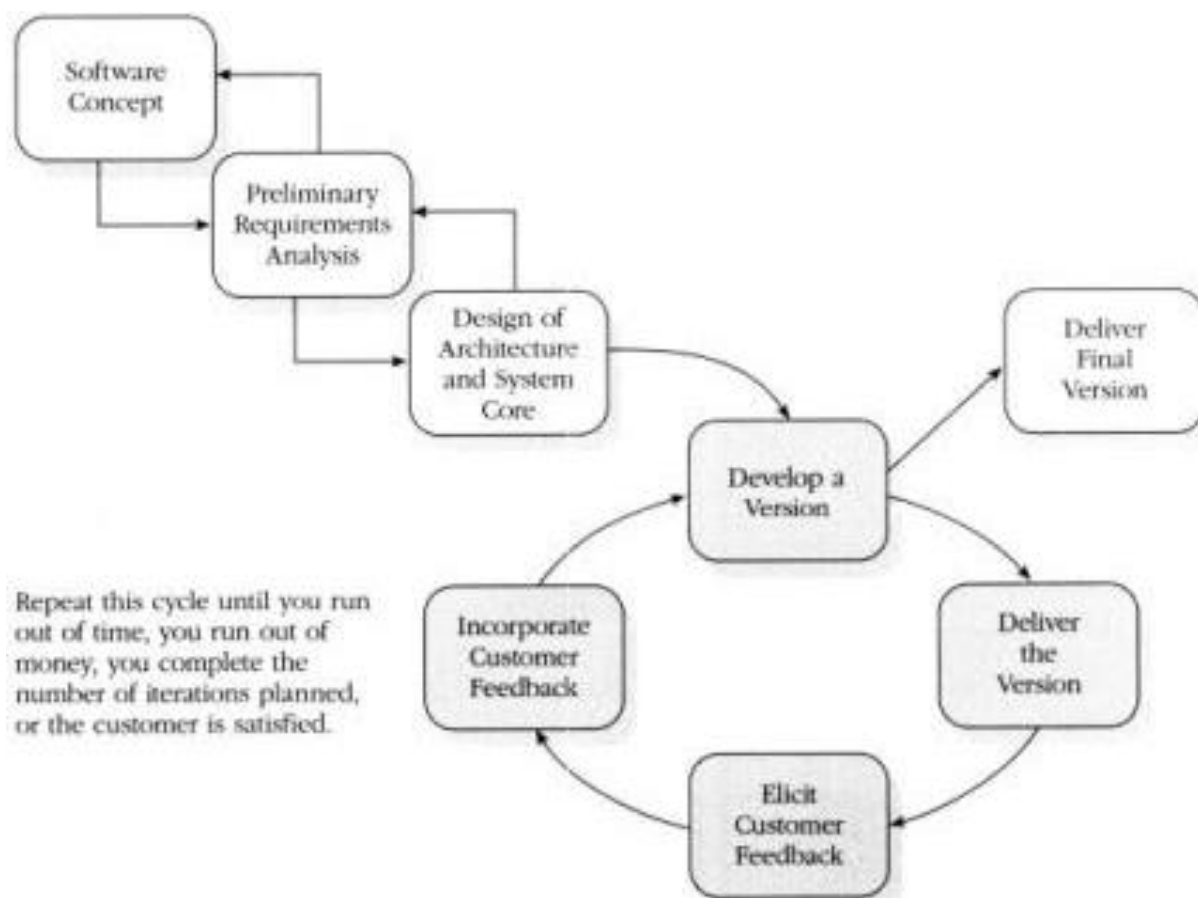
• 缺点

- ❑ 领导对于许多技术细节是外行
- ❑ 领导未必懂软件项目的管理
- ❑ 行政命令未必能管好软件团队
- ❑ 领导精力有限，领导很忙时，团队怎么办？

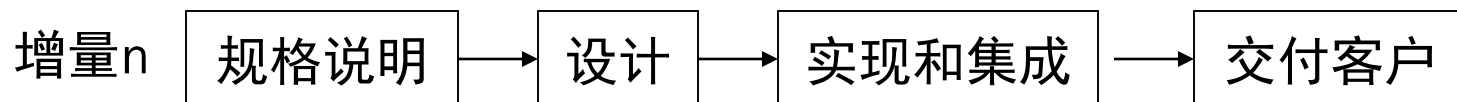
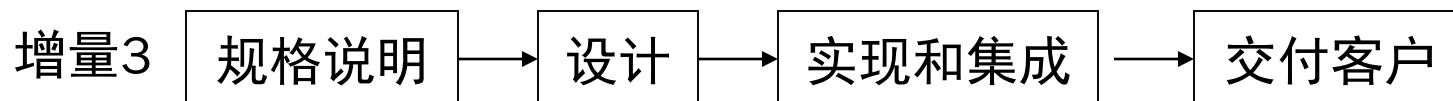
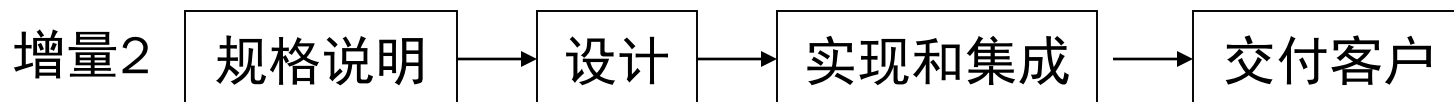
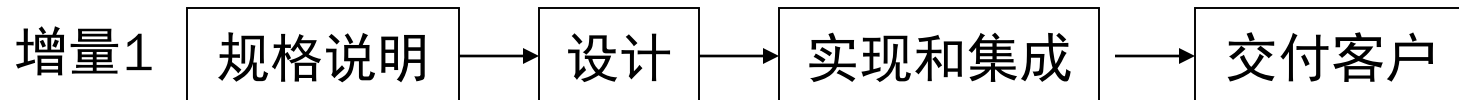
渐进交付模型

- 演进式原型和分阶段交付
- 重复演进循环直到

- 时间到了
- 钱花光了
- 用户满意了



增量模型



- **增量**

- 小而可用的软件

- **特点**

- 在前面增量的基础上开发后面的增量
 - 每个增量的开发可用瀑布或快速原型模型
 - 迭代的思路

- **MVP(Minimal Viable Product)**

- 最小可行产品，又称最小功能集
- 把产品最核心的功能用最小的成本实现出来，然后快速征求用户意见

- **MBP(Maximal Beautiful Product)**

- 最强最美产品
- iPhone、iPad

你们的项目将选择什么模型？

- 模型很多...
- 但时间有限



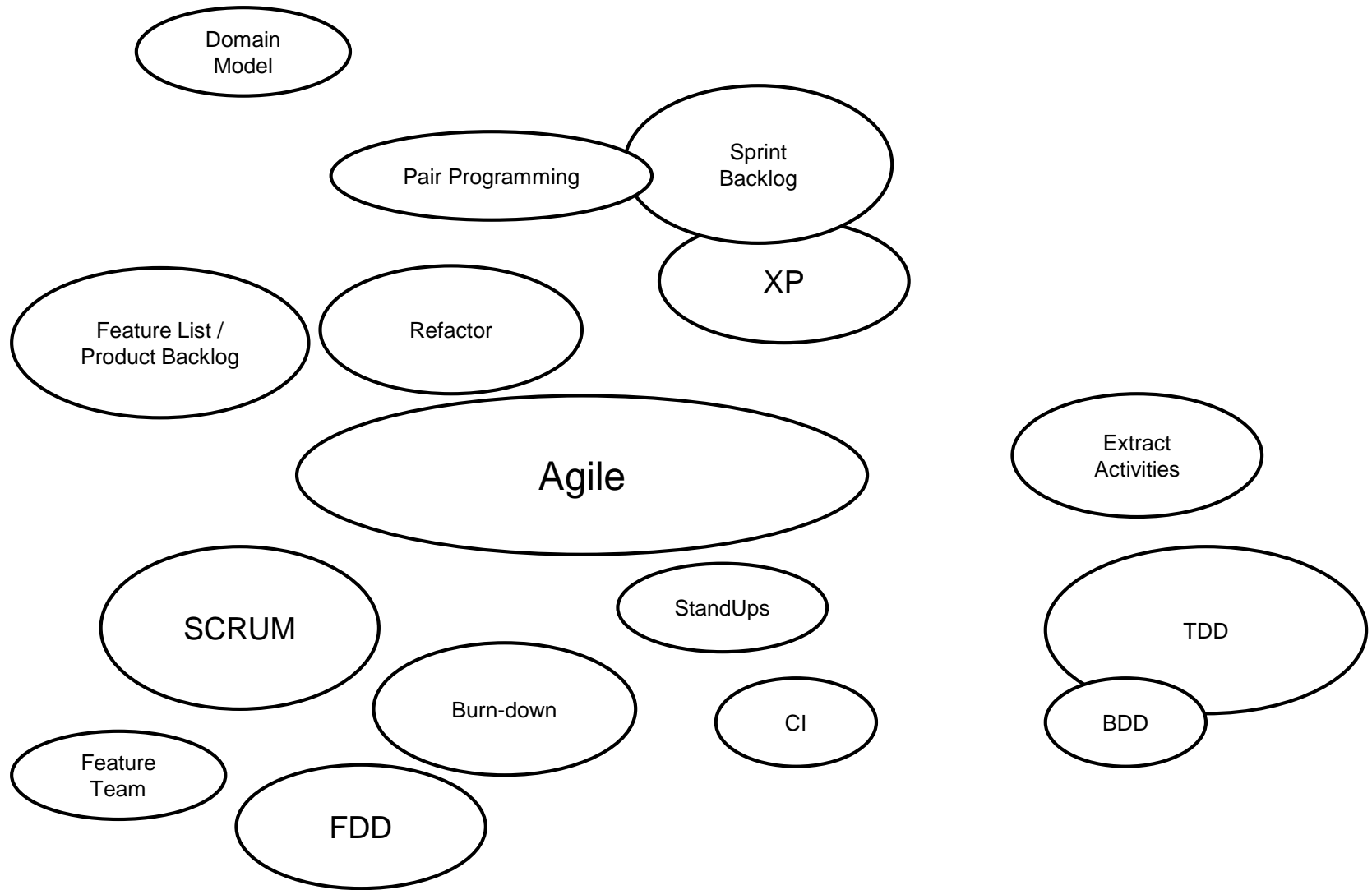
软件开发环境国家重点实验室
State Key Laboratory of Software Development Environment

敏捷开发流程

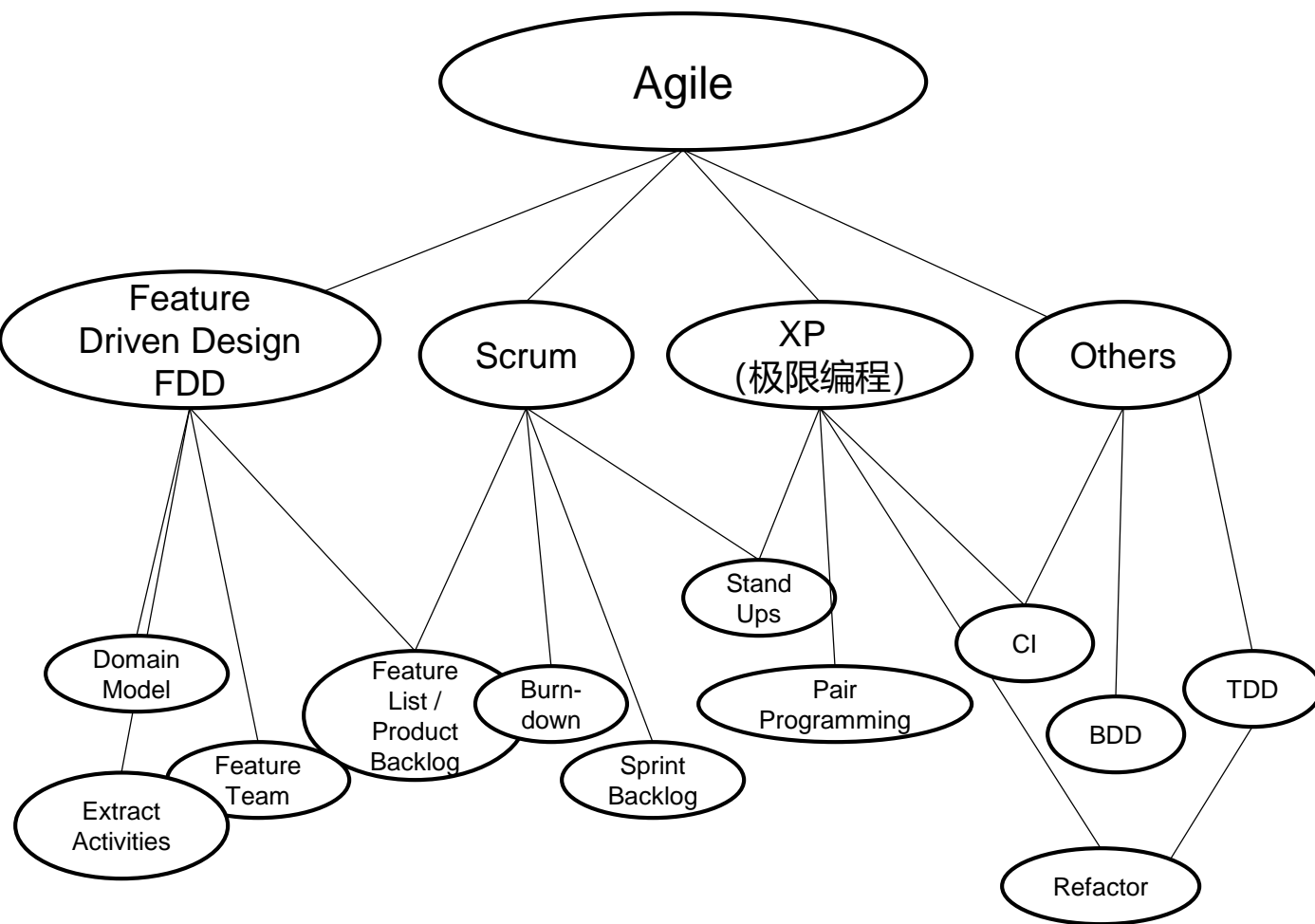
- **“敏捷开发流程” 是一系列价值观和方法论的集合**
 - ▣ 始于2001年

现有的做法	敏捷的做法
流程和工具	个人和交流
完备的文档	可用的软件
为合同谈判	与客户交流
执行原定计划	响应变化

什么是敏捷?



什么是敏捷?



敏捷是一种思潮.....

它涵盖了一组软件开发的方法论.....

这些方法论又是建立在许多最佳实践方法上的

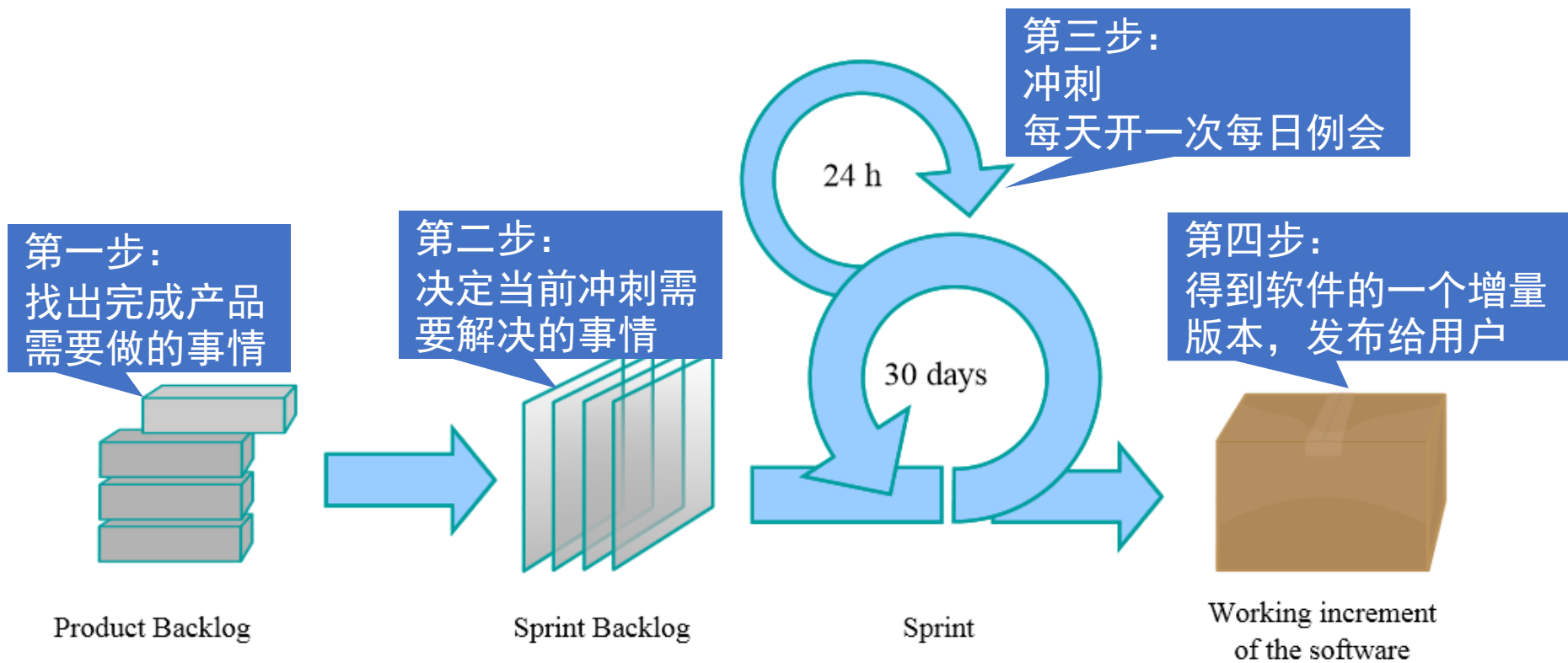
- **尽早并持续地交付有价值的软件以满足顾客需求**
- **敏捷流程欢迎需求的变化，并利用这种变化来提高用户的竞争优势**
- **经常发布可用的软件，发布间隔可以从几周到几个月，能短则短**

- **业务人员和开发人员在项目开发过程中应该每天共同工作**
- **以有进取心的人为项目核心，充分支持信任他们**
- **无论团队内外，面对面的交流始终是最有效的沟通方式**

- **可用的软件是衡量项目进展的主要指标**
- **敏捷流程应能保持可持续的发展。领导、团队和用户应该能按照目前步调持续合作下去**
- **只有不断关注技术和设计才能越来越敏捷**

- **保持简明——尽可能简化工作量的技艺——极为重要。**
- **只有能自我管理的团队才能创造优秀的架构、需求和设计**
- **时时总结如何提高团队效率、并付诸行动**

- **SCRUM是敏捷软件开发中常见的一种迭代、增量式的项目管理方法论**

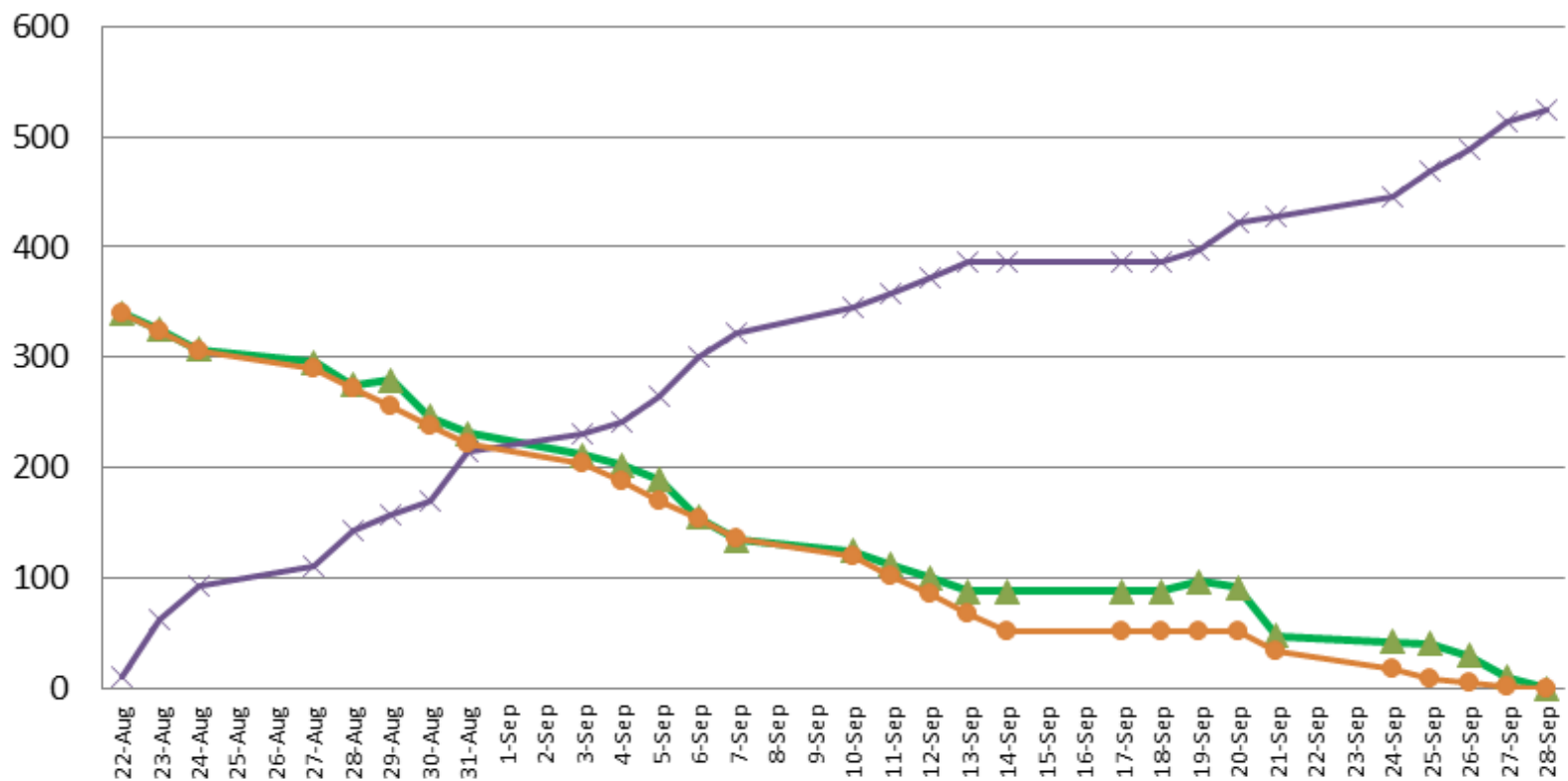





- **时间驱动(Time-boxed)**
 - 15分钟的站立会议
 - 固定时间长度的冲刺，避免拖延
- **冲刺阶段不能有需求变更，等到冲刺结束之后再进行**
- **每一次冲刺结束之后，需要总结上一次的经验教训，争取下一次做的更好**

燃尽图

Dev Work

Proj
v1.1
burn
down
chart



	22-Aug	23-Aug	24-Aug	27-Aug	28-Aug	29-Aug	30-Aug	31-Aug	3-Sep	4-Sep	5-Sep	6-Sep	7-Sep	10-Sep	11-Sep	12-Sep	13-Sep	14-Sep	17-Sep	18-Sep	19-Sep	20-Sep	21-Sep	24-Sep	25-Sep	26-Sep	27-Sep	28-Sep
 remain hour	340	325	307	295	275	279	246	231	211	202	189	155	134	124	112	100	87	87	87	87	96	91	47	42	40	29	9	0
 complete hour	9	61	93	111	143	157	169	215	230	242	264	301	322	346	358	372	386	386	386	386	397	422	427	445	468	488	513	524
 projected remaining hour	340	323	306	289	272	255	238	221	204	187	170	153	136	119	102	85	68	51	51	51	51	51	34	17	8	4	1	0

看板图



Backlog	Analysis	Development	Testing	Done
Waiting for Analysis	Analysis	Waiting for Dev	Dev	Waiting for Testing
Print Receipt	Add Local Sales Tax	Purchase optional extra item	Sign up at website	
Refund				

Planned Queue WORK	SLOT 1	SLOT 2	ACCEPT Work to be Accepted	DONE Completed Work
<p>① Andrew B. Confirm that it's ready to be deployed to:</p> <ul style="list-style-type: none"> 1.8 - properly supported - ready for monitoring Good make to make it live. 2014-05-10 		<p>② Karl Use status for:</p> <ul style="list-style-type: none"> - copy data - main data. Prev vs 	<p>③ Karl</p> <p>Filed</p>	

BACKLOG	NEXT	NEXT(10)	NEXT(4)	DOING(3)	WAITING(2)	DONE
<p>Capacity vs Capability</p> <p>Double loop learning</p> <p>SHR</p>	<p>paper backlog</p> <p>Ignite Seattle</p> <p>Business force 5</p> <p>Kanban board</p>	<p>Game updates</p> <p>LKBE 11 DJAA PRES</p> <p>ppt J curve</p> <p>Company Miro agenda</p>	<p>BACK UPS</p> <p>KANBAN TOOLS RESEARCH</p> <p>KWR NOV 8th Browser mac</p> <p>Project Education</p>	<p>MARVEL OF KANBAN Adaptability</p> <p>Website mock-ups review</p> <p>CONTINUOUS DELIVERY Chap 11</p>	<p>TRAINING CURRICULA UPDATES</p> <p>Interrupts</p>	

敏捷的团队

- **自主管理**

- 自己挑选任务
- 每次冲刺结束之后总结不足，提出改进并实施

- **自我组织**

- 团队成员要联合起来对项目负责
- 有人落后要帮他改进

- **多功能型**

- 每个人全面负责规格说明书、沟通和测试等

• 项目的期限不能改动；敏捷能帮我早日完成任务？

- ❑ 回答：敏捷不是万能。敏捷的方法能帮助你更早地知道你是否能如期完成任务。仅此而已。
- ❑ 敏捷的方法（迭代的方式）能帮你尽快让用户看到项目的（部分）价值。
- ❑ 当你尽早交付（部分）价值的时候，也许用户对你目前交付的东西已经很满意了，这样你就不用再花时间来实现在其他计划中的事情。
- ❑ 另一种可能是，用户看到了（部分）系统，他们对整个需求有了新的认识，这样你就可以实现他们新的需求，而不用再浪费时间。



- **敏捷宣言表明的一些优先级，不要把敏捷变为教条！**
- **要让一线团队成员做决定**

敏捷的适用范围



最适用方式 客观因素	敏捷	计划驱动	形式化方法
产品可靠性要求	不高，容忍经常出错	必须有较高的可靠性	有极高的可靠性和质量要求
需求变化	经常变化	不经常变化	固定的需求，需求可以建模
团队人员数量	不多	较多	不多
人员经验	有资深程序员带队	以中层技术人员为主	资深专家
公司文化	鼓励变化，行业充满变数	崇尚秩序，按时交付	精益求精
实际的例子	写一个微博网站	开发下一版本的办公软件；给商业用户开发软件	开发底层正则表达式解析模块；科学计算；复杂系统的核心组件

- 要全面的理解敏捷原则，可以阅读Martin Fowler的文章

<http://martinfowler.com/agile.html>

- 软件开发：敏捷开发模式，无论是产品还是运营都要懂

<https://www.toutiao.com/a6671886794155360771>





- **一、趣学**
- **二、高校教师成果管理系统**
- **三、基于科研社交的微知识发布与分享系统**
- **四、公众科学项目发布与数据采集系统**
- **五、体育赛事资讯平台**
- **六、北航大学生活动发布与社交平台**
- **七、自拟题目**



选题一：趣学

• 背景

- 无论是在家还是在学校，当今的大学生都面临着如何高效学习的问题。尤其是疫情当前，如何在家中高效学习成为了很多大学生的一大难题。基于这种需求，本项目计划开发“趣学”APP，让高效效率，找到拥有相同学习目标的小伙伴不再困难，高效学习，就“趣学”！

• 目标

- 为了提高大学生学习效率，提高学习氛围，本项目计划开发一款面向大学生的群组学习软件，用户可以方便的找到拥有相同学习目标的人，实现共同学习，共同进步。主要实现群组聊天，资料分享，知识分享等学习功能，针对特定课程的聊天机器人等功能。



选题一：趣学

• 主要功能需求如下：

- ❑ 管理端需求（Web端）：实现预置管理员账号登录功能，用户管理，群组管理，接入北航教务系统，并根据教务系统的课程信息创建学习群组，以及基本的消息推送功能
- ❑ 用户端需求（移动设备：iOS 或 Android，可选：Web 端）：实现注册、登录功能；从北航教务系统获取个人课表并以合适的方式展示；用户可以加入官方的课程群组，也可以自行创建群组，群组可以实现基础的聊天，资料分享等功能；群组的知识分享功能，在群组中实现知识分享模块，任何群组中的成员均可以在知识分享模块中发表个人的学习心得，知识总结等内容，任何群组的成员均可以查看知识分享模块中的内容，并可以点赞，评论，收藏；实现个人时间管理功能，包括但不限于日程任务的创建，to-do list，番茄钟等学习功能。
- ❑ AI 模块：在官方群组的功能中，加入聊天机器人功能，可以通过预设字段，实现快速查看课程信息，分组名单，作业题目等课程信息。管理端应该可以方便的设置预设字段，设置各种查询信息。也可以根据机器学习，深度学习等方法自行添加功能，实现更多有趣的功能。





• 背景

- 众多高校为了收集并保存全校师生的科研学术成果，开发了一些教师成果系统，如北航教师成果数据库、清华学者库、西安交通大学机构知识门户等，都提供了一个全面收集了全校智力成果的文献资料库，为全校师生提供成果存档、管理、发布、检索和开放存取，向全球范围提供免费和永久的访问。

• 目标

- 本项目计划基于某一高校教师成果数据库中的已有数据，进行爬取构建高校学者库及其成果库，开发一个高校教师成果管理系统，同时对现有教师成果库的功能与数据进行扩充。主要实现数据采集，学者认领，学者与成果管理，学者画像，导师推荐等功能。



• 具体需求

- ❑ 数据采集需求：需要基于现有高校的教师成果库，完成院系机构的采集，学者基本信息的采集，论文，书籍专利等成果的采集，以及学者与成果信息之间关联信息的构建，同时基于学者的成果信息分析学者的研究领域与兴趣及其他学术画像信息。
- ❑ 管理端需求(Web)：需要实现预置管理员账号登录功能，学者管理，成果管理，学者认证审批，学者提交成果审批，院系机构管理，数据爬取任务管理,相关数据统计并展示等功能。
- ❑ 用户需求(小程序或者Android或者iOS)：实现用户浏览学者及成果信息等功能，以及用户输入感兴趣的研究方向后，按照学者的研究领域相关性以及学术影响力等信息为用户推荐导师(需采用AI算法实现)。



选题三：基于科研社交的微知识发布与分享系统

• 背景

- 2019年3月，美国防高级研究计划局（DARPA）正式推出“聚网”（Polyplexus）新型社交媒体平台，旨在利用其实时交互能力加速前沿科技领域新思想、新假设、新概念的形成与论证，快速将新创意转变为高价值解决方案，加速协同创新科技研发进程。
- “聚网”平台目的是聚集不同学科、不同背景的个人和团队提出的构想与见解。聚网具有一个信息源公开发布区，用户可发布类似推特的“微公告”（micropub）。“微公告”包括“证据微公告”（evidence micropub）和“猜想微公告”（conjecture micropub）两种，前者旨在向公众推广期刊、书籍、研究论文等来源的新科学发现，后者则基于两条不同的“证据微公告”提出新的、合理的科学猜想。

• 目标

- 对于国内广大的科研人员来说，国内目前缺少一个用户科研信息分享、传播以及获取的网络社交平台。本系统目的是建立基于科研社交的微知识发布与分享系统：主要目的是科研从业者针对科学问题，通过文献阅读，积累科学事实（微证据evidence），提出科学猜想（微猜想conjecture），并形成针对微证据和微猜想的分享系统，从而搭建一个以微知识为交流介质的网络社交媒体平台。该系统类似于科技界的微博，通过科研人员收集科学事实，提出科学猜想来促进不同领域的科研学者和行业专家之间的知识流通和交流，推动科技创新的发展

• 具体需求

- **微知识发布：**微证据和微猜想的发布
- **微知识评审：**
- **社交功能：**用户的注册登录。浏览的微知识可以实时更新，可以关注其他用户并且获得用户的最新动态。对微知识的点赞评论收藏，用户可以向论坛中特定人发送内部消息。使用AI算法，根据标签等信息实现智能推荐，让用户更容易看到对其有用和感兴趣的微知识。实现消息管理，包括系统通知和用户消息通知。



选题四：公众科学项目发布与数据采集系统

• 背景

- 公众科学(citizen science)也称公众参与式科学研究, 指包含了非职业科学家、科学爱好者和志愿者参与的科研活动其范围涵盖科学问题探索、新技术发展、数据收集与分析等。
- 在著名公众科学网站 (citsci.org /citizenscience.org) 注册的生态学和环境科学相关领域的项目已超过1K个, 研究内容涉及动植物监测、入侵种调查、大气质量调查、水质调查、气候变化监测等。
- 而中国公众科学目前处于起步阶段, 国内大众和学者对于公众科学的了解还较少。但已有一些有益的尝试, 例如国家标本平台植物标本馆在线信息系统, 建设了植物多样性调查在线协作平台www.cfh.ac.cn, 已成为中国最大的生物多样性调查公众科学协作体系和数据积累中心。



选题四：公众科学项目发布与数据采集系统

• 目标

- 研发中国公众科学项目发布与数据采集系统，为科学家提供项目发布、数据收集、统计分析与可视化等功能，实现项目的快速搭建与发布、数据实时采集与分析、后台数据审核与管理。为项目志愿者提供便捷的数据采集工具（App），能够实现志愿者采集数据的快速录入和项目查询等功能；也可提供科学家与志愿者的线上沟通交流功能。
- 系统包括Web端和App端两个部分，其中Web端包括项目发布与管理、数据录入功能；APP端主要支持数据录入，提供数据的智能处理与标记等。

• 具体要求

- 管理员需求：实现用户管理及认证（共分管理员，科学家、助理、志愿者四种角色，其中科学家与助理需要认证），项目管理等功能。
- 科学家与助理需求：实现项目发布与助理设置(科学家权限)，设置数据录入要求，设置数据统计指标，用户加入项目确认(若需要)，数据管理(查看与审核)，导出数据。
- 用户端需求(APP)：实现未注册用户可以浏览网站基本信息，浏览项目与数据。实现用户注册与登录，身份认证申请，项目加入申请，数据采集与录入，采集任务通知。



• 背景

- 目前各类体育赛事都有其官方资讯发布网站以及第三方新闻媒体等数据源头，且随着体育项目的扩展以及赛事的发展，体育赛事已经成为了爱好者们最关注的话题看点之一。

• 目标

- 为了方便用户关注即将到来的赛事和自己喜爱球队的赛程，以及浏览之前进行的赛事详情，本项目计划面向某类竞赛项目（可自选），基于目前可公开访问的赛事数据资讯源头，开发一款体育赛事资讯平台。本项目应实现数据采集，赛事列表查看，赛事数据详情查看，发表评论，比赛结果预测等功能。



• 具体需求

- ❑ 数据准备
- ❑ 管理端需求(Web): 需要实现管理员登录、赛事数据库管理、赛事内容管理、录入新赛事信息、用户管理以及留言管理。
- ❑ 用户端需求(Android或者iOS APP): 需要实现用户浏览赛事列表（已结束赛事与未来赛事）、赛事战报信息与相关图片查看、未来赛事预告提醒、针对某场赛事评论留言与回复评论以及查看系统提供的赛事结果预测。
- ❑ AI预测: 系统根据自己的预测算法结合过往比赛数据，对未来的比赛进行胜率（或比分）预测，为用户提供参考。



选题六：北航大学生活动发布与社交平台

• 背景

- 目前北航的学术讲座、博雅课堂、社团活动和各个学生会组织的活动并没有一个统一的发布平台和入口，学生、老师及活动组织者需要通过不同的网站进行注册报名，并且这些网站也没有相关的日程提醒功能，也无法让活动参与者对组织者提供反馈，我们旨在用一个带有网页端的平台，来改变这一现状，方便北航师生的课余生活。

• 目标

- 该系统为面向北航全校师生的活动发布、管理和社交平台，旨在方便全校的活动组织者和参与者，在活动发布、宣传通知、日程提醒和参与层面，给予一个统一的发布平台，并基于推荐算法为广大师生提供当前北航正在进行的、人气高的或用户感兴趣的活动。

• 具体需求

- ❑ 管理端需求(Web): 需要实现管理员登录、审核注册用户、审核与批准活动、管理论坛信息等功能。
- ❑ 用户需求(Android或者iOS APP): 需要实现用户注册登录以及信息编辑。用户可以浏览当前全部开放的活动, 或按照关键词搜索相关活动, 并申请加入该活动; 用户也可以以活动发起者的身份发起自己的活动, 发起后应可以修改活动相关信息; 基于已经参加的活动, 用户端应以日程表的形式为用户展示即将到来的活动, 并及时提醒; 用户也可以对自己参加过的活动进行评价。
- ❑ 个性化推荐: 需要设计并实现一款推荐模型, 将用户可能感兴趣的热门活动推荐给用户。



选题七：团队自拟

- ❑ 团队选择并确定感兴趣的题目，并与助教和任课教师提前进行沟通，确认后可作为软件工程团队项目进行开发
- ❑ 基本要求：项目需求必须来自实际的业务，项目规模应与前六个选题规模相当。
- ❑ 系统应涉及人工智能的一个或多个相关算法



- **组队：确定分组**
- **每个项目最多2个团队选择，报名方式将在周五微信群公布，先到先得**
- **5人团队只能选择高校教师成果管理系统、基于科研社交的微知识发布与分享系统其中之一**
- **团队项目详细文档全部发布在课程中心**
- **团队项目的部署和过程文档需全部上传华为软开云，每个团队一台虚拟机，助教会统一分配**