



CS2001

软件工程

6. 软件过程与团队合作

— 软件过程与敏捷开发

软件开发流程



西乔

设计师, 项目经理,
2006年起携创业团队从事Web
技术外包开发及产品咨询顾问。

如果你有什么好玩的关于程序员的故事、对话、代码,
愿意通过漫画的形式分享,
请给西乔发邮件: arthur369@gmail.com



作者:
西乔

设计师, 项目经理, 06年起携创业团队从事Web技术外包开发及产品咨询顾问。
如果你有什么好玩的关于程序员的故事、对话、代码, 愿意通过漫画的形式分享,
请给西乔发邮件: arthur369@gmail.com



类比：饮料生产流程

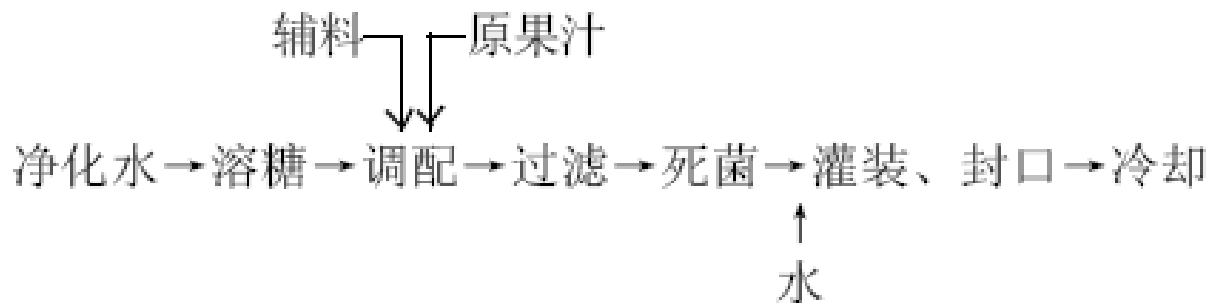
由一系列步骤和顺序组成

工艺流程

火棘原汁的提取工艺流程

采果→挑选→清洗→软化打浆→酶解→分离、澄清
过滤→杀菌→冷却→原汁保存

火棘果汁饮料的加工工艺流程



类比：饮料生产流程

有一些关键质量控制点及相应的手段和措施

火棘饮料生产中 HACCP 体系的关键控制点及控制方法

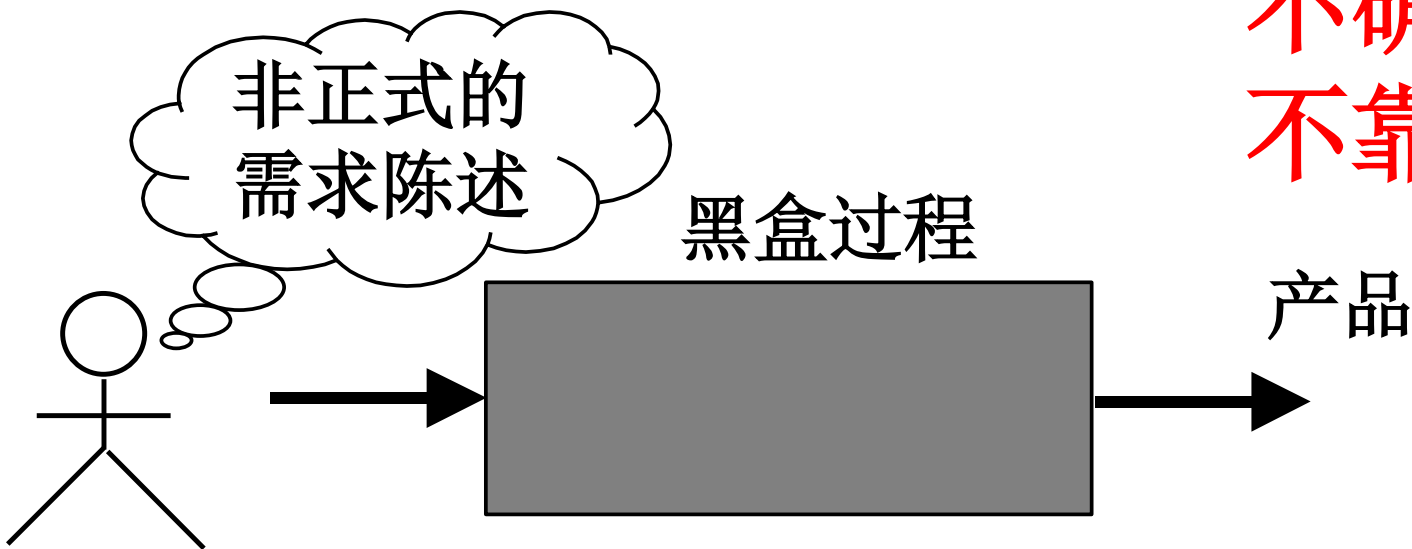
Critical points and controlling methods of *Pyracantha fortuneana* beverage in HACCP system

工艺环节	关键控制点	控制范围	监测方法	控制手段	校正措施
原料	成熟度、新鲜度、虫果及腐烂率	色泽鲜艳，充分成熟，无病虫害烂果，干缩果及枝叶杂质	感官检验	杜绝收购未成熟、质量差的原料	有选择地进行原料收购
软化打浆、酶解	料水比、酶用量和酶解时间	料水比 1:2，酶用量 0.2%，6h	量具和时钟测量，酒精实验	严格按工艺参数进行操作	根据酒精实验结果对酶用量及酶解时间进行微调
分离、澄清、过滤	设备卫生状况、明胶用量、络合时间	设备干净卫生，明胶配成 3%~5% 的溶液，络合 20~24h	量具和时钟测量，感官检验	设备定期进行 CIP 清洗、严格按工艺参数进行操作	根据每批小试结果调整明胶用量及络合时间
杀菌、贮存	设备、贮桶卫生状况，灭菌温度及时间，原汁贮存温度及贮存时间	设备、贮桶干净卫生，115 ± 2℃ 下保持 3~5s，阴凉干燥处存放	感官检验，微生物分析检测	设备定期进行 CIP 清洗、严格按工艺参数进行操作	根据微生物分析检测结果调整工艺参数
配料	原辅料质量	符合相关卫生质量标准	原汁进行理化及卫生质量检测	使用符合工艺要求的原辅材料，严格控制辅助材料的进货渠道	有选择性地对辅料购买
过滤、灭菌	设备卫生状况，灭菌温度及时间	设备干净卫生，120 ± 2℃ 下保持 3~5s	感官检验，微生物分析检测	设备定期进行 CIP 清洗，严格按工艺参数进行操作	根据分析检测结果调整工艺参数
罐装、封口	设备及包装物卫生状况	设备干净卫生，70~80℃ 热灌装		设备定期进行 CIP 清洗，严格按工艺参数进行操作	根据分析检测结果调整工艺参数

HACCP (hazard analysis and critical control point)
是国际上共同认可和接受的食品安全保证体系

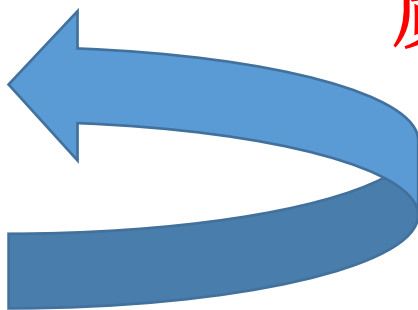
如果没有软件过程...

不确定！
不靠谱！



需求不确定、不完整

质量？工期？成本？



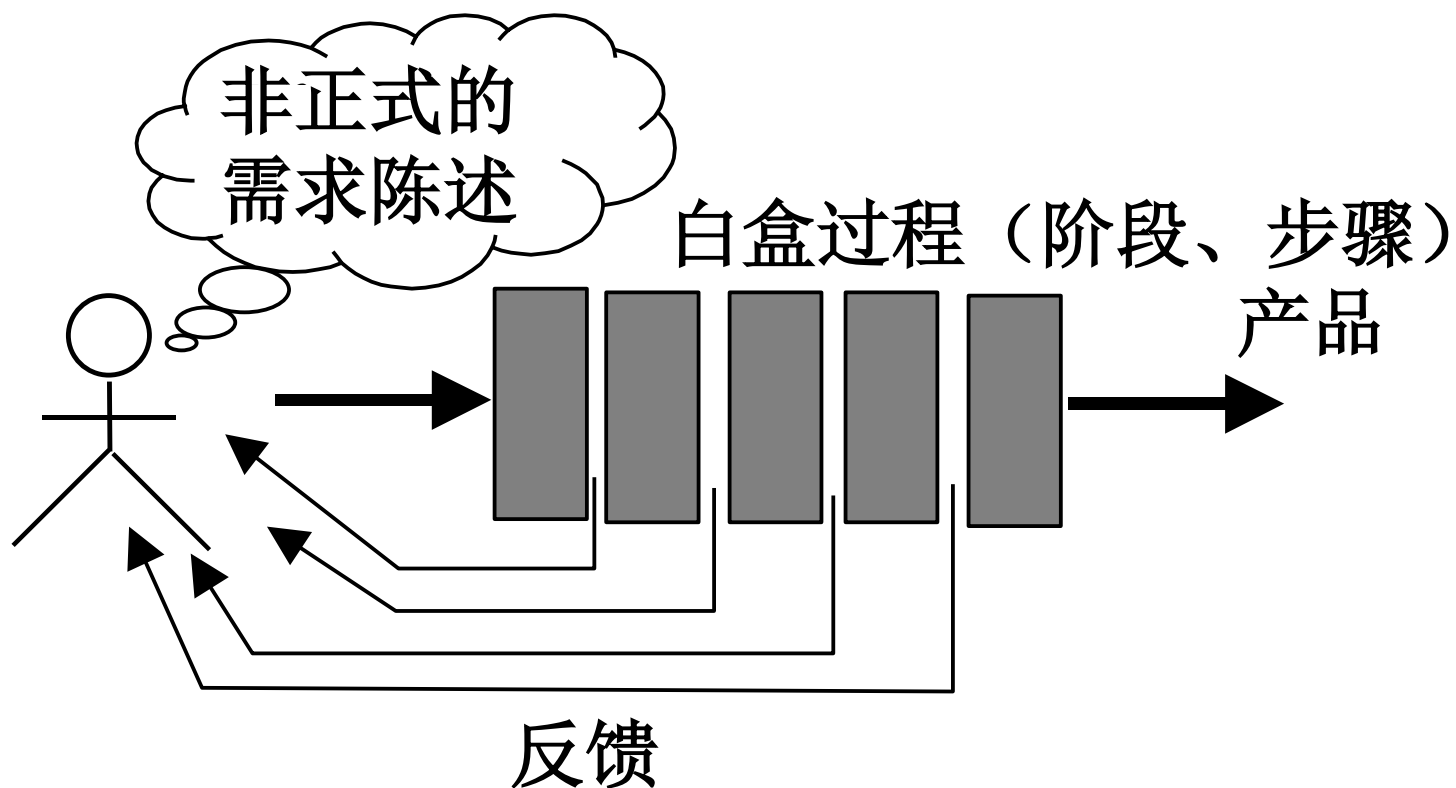
Code & Fix (先写再改)

软件过程：让开发过程可见、可控

遵循成功经验

中间过程（进度、质量）可见

根据反馈及时调整



基本的软件开发活动

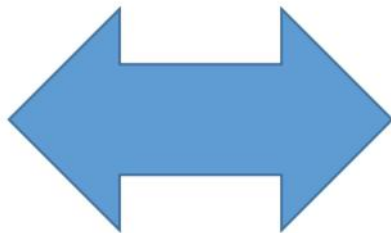
- 规格说明（需求）：定义待开发的软件系统什么样、应该做什么
- 设计和实现：定义软件系统的内部结构并进行实现
- 确认（测试）：检查所实现的系统是否是客户和用户想要的
- 演化（维护）：根据客户需求及环境变化等不断对软件进行修改

不同的开发过程按照不同的顺序对这些活动进行编排

软件过程：计划驱动 vs. 敏捷

适应变化
快速迭代
个体与交互
可运行的代码

敏捷



计划

遵循计划
按部就班
过程与工具
详尽的文档

©Kris
Modern
Analyst

充分考虑软件的特点
灵活、便于调整

传统工程化思维
稳定、便于协调



The ways of working out our differences took a sudden dangerous turn.

没有好与坏，只有合适不合适

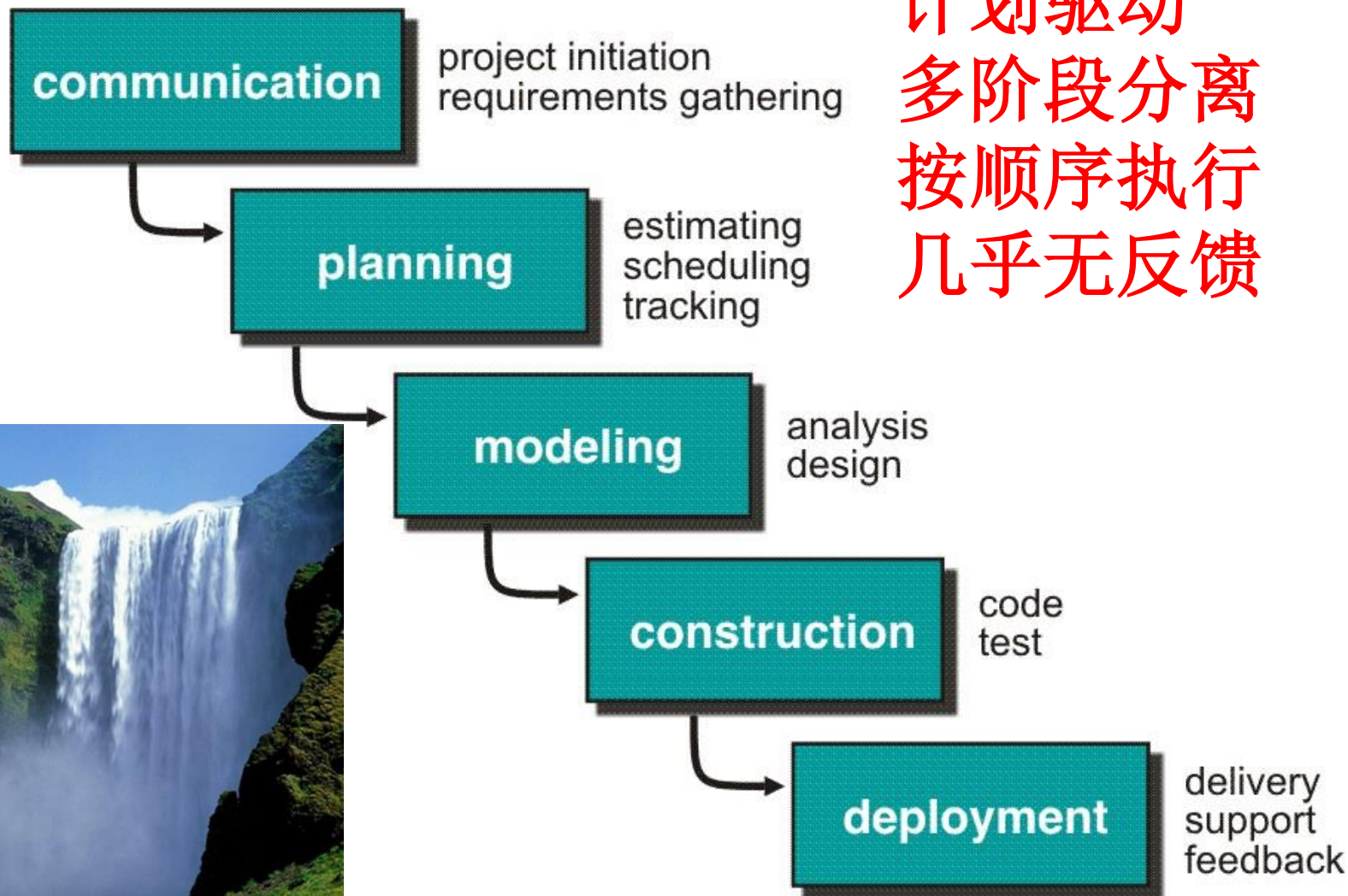
- **计划驱动的过程：所有过程活动均事先计划，按照计划衡量进度**
 - ✓ 需求可以事先清楚、完整定义，且不会有太大变化
 - ✓ 最终得到的软件系统有较高的可靠性、安全性要求
 - ✓ 涉及多个部门或团队，涉及复杂的协调和沟通
- **敏捷开发：只做增量的短期计划并根据变化和反馈不断进行调整**
 - ✓ 需求不确定且经常变化（变化蕴涵着竞争优势）
 - ✓ 对于最终软件系统的可靠性和安全性要求没那么高
 - ✓ 开发团队规模较小且能够充分沟通

在实践中二者经常会有所结合

经典过程模型

- 瀑布模型
- 增量开发模型
- 螺旋模型
- 统一过程模型

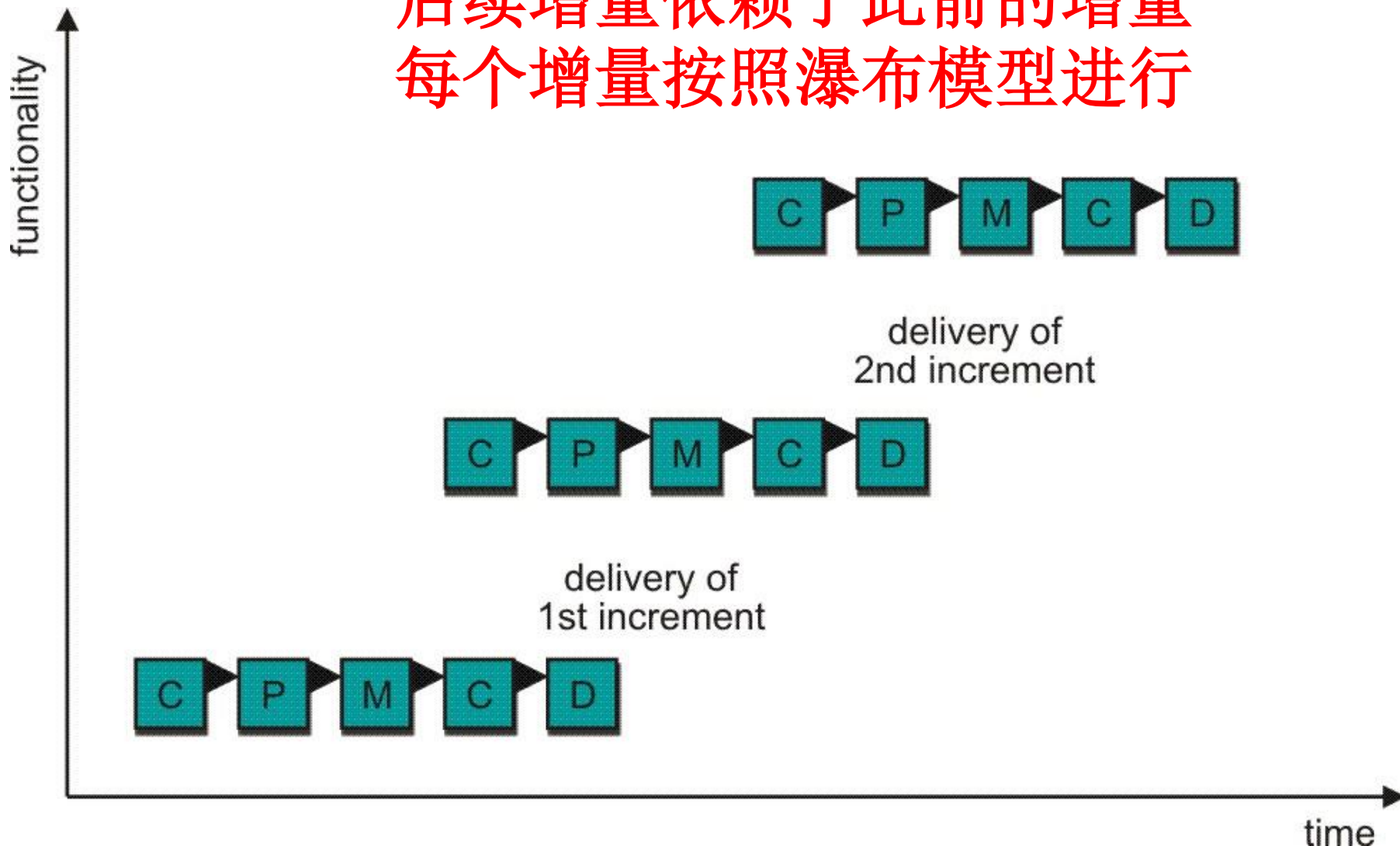
瀑布模型



计划驱动
多阶段分离
按顺序执行
几乎无反馈

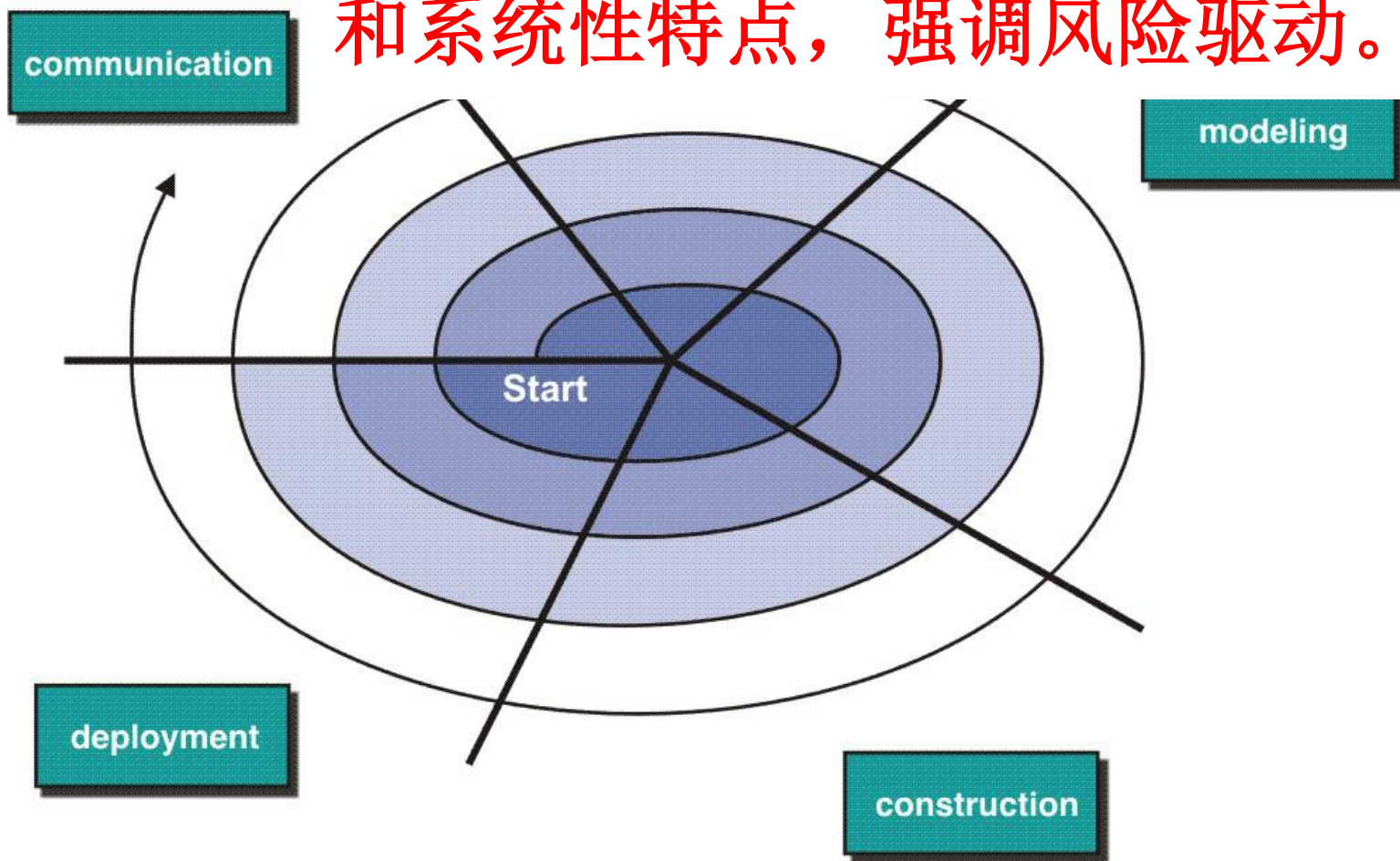
增量开发模型

事先计划多个增量
后续增量依赖于此前的增量
每个增量按照瀑布模型进行

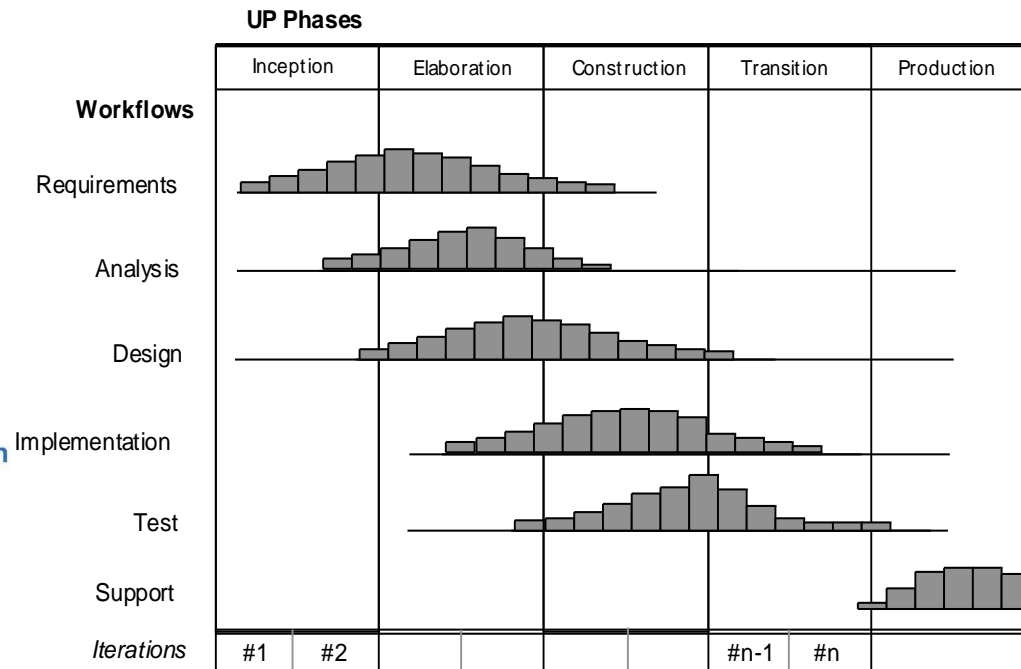
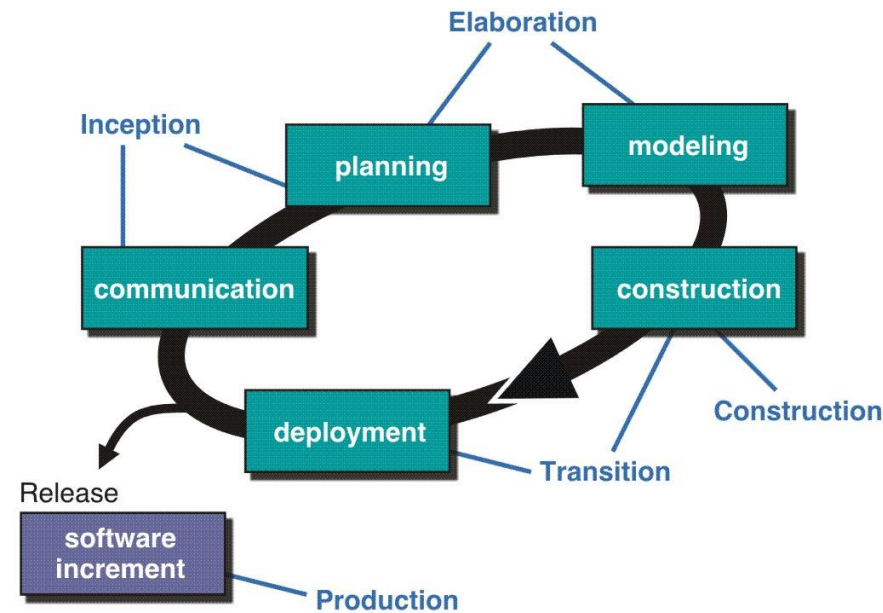


螺旋模型

演化式软件过程模型，结合原型的迭代性质和瀑布模型的可控性和系统性特点，强调风险驱动。



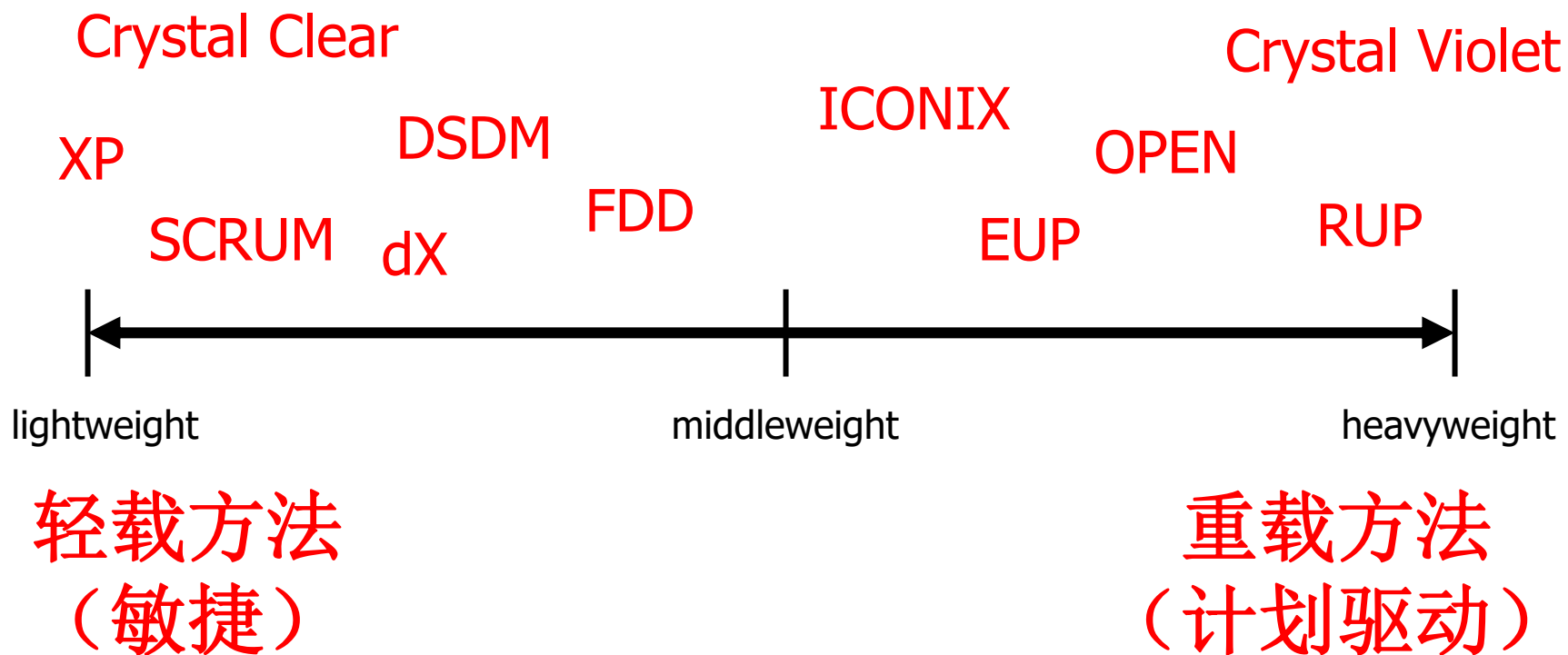
统一过程模型 (Unified Process)



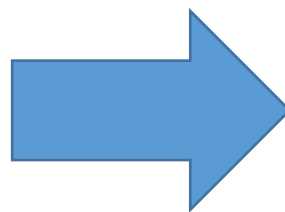
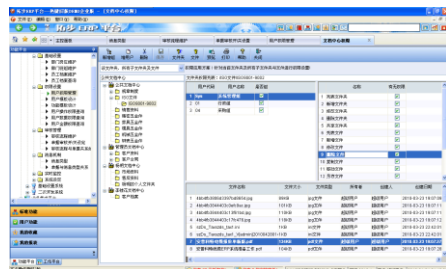
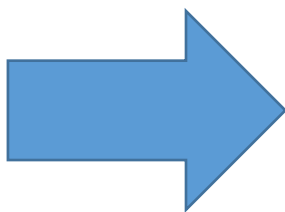
预定义多个开发阶段
每个阶段包含多次迭代
每次迭代都包含多个开发活动（强度各不相同）

用例驱动，以架构为核心，迭代并且增量

软件过程频谱



敏捷产生的背景



大型机时代
(科学计算、军工系统等)

桌面计算时代
(信息管理系统、办公软件等)

互联网时代
(电子商务系统、社交与支付App等)

应用面窄，需求确定，稳定性要求高，更新慢

应用面有所拓展，需求阶段性发生变化，通过光盘等发布载体更新

应用直面消费者，快速更新且竞争激烈，快速变化、随时更新

敏捷宣言

“We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:



- *Individuals and interactions* over processes and tools
- *Working software* over comprehensive documentation
- *Customer collaboration* over contract negotiation
- *Responding to change* over following a plan

That is, while there is value in the items on the right, we value the items on the left more.”

-- Kent Beck et al.

On February 11-13, 2001, at The Lodge at Snowbird ski resort in the Wasatch mountains of Utah, seventeen people met to talk, ski, relax, and try to find common ground and of course, to eat. What emerged was the Agile Software Development Manifesto. Representatives from Extreme Programming, SCRUM, DSDM, Adaptive Software Development, Crystal, Feature-Driven Development, Pragmatic Programming, and others sympathetic to the need for an alternative to documentation driven, heavyweight software development processes convened.

敏捷过程的特点

- 客户需求（场景、故事）驱动
- 认识到有效的计划都是短期的
- 迭代地开发软件同时强调构造活动（设计与实现交织）
- 频繁交付可运行的“软件增量”
- 根据变化进行适应性调整

强调适应而非预测

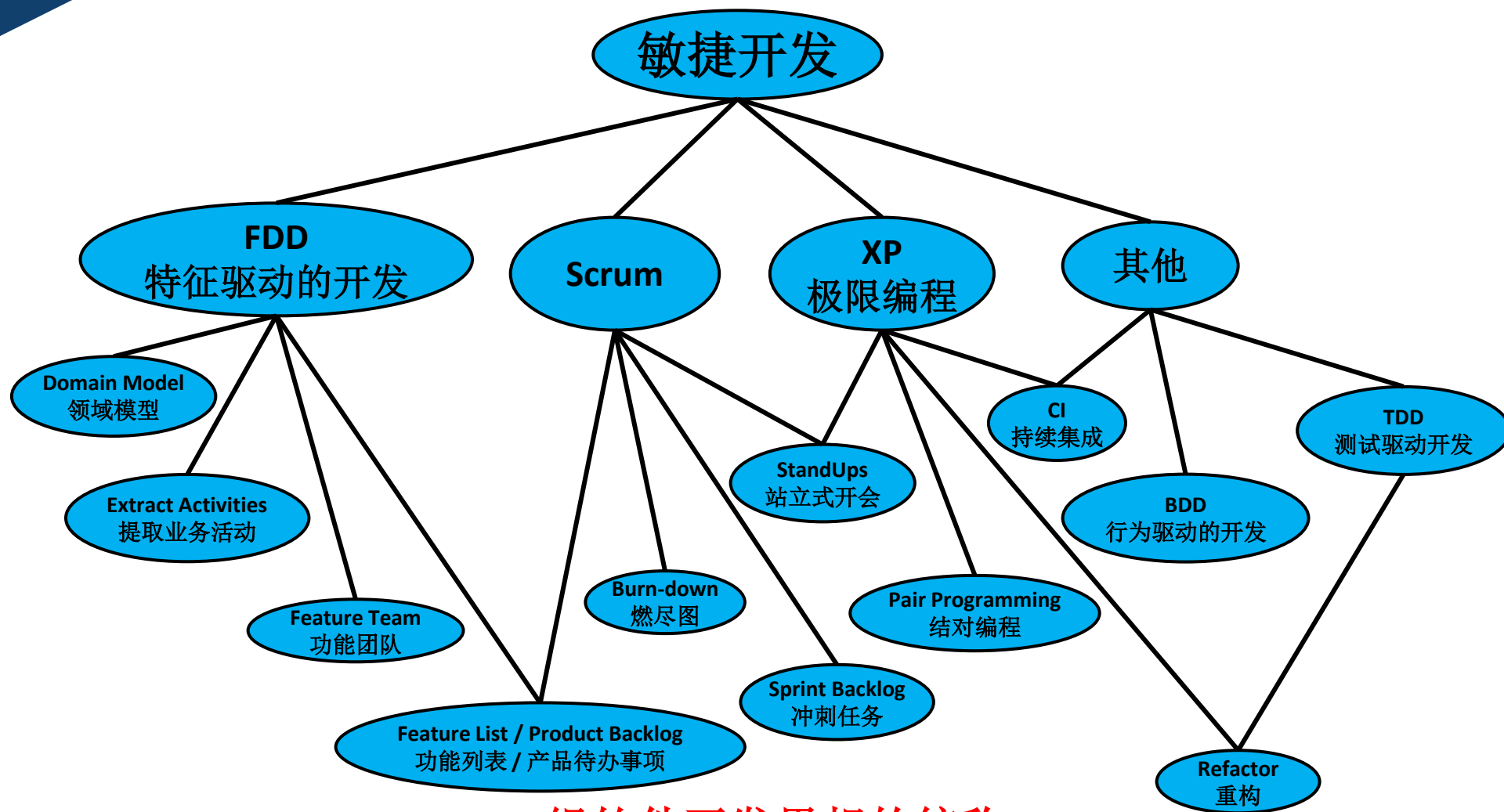
敏捷开发原则-1

- 尽早并持续地交付有价值的软件
- 经常发布可用的软件，发布间隔从几周到几个月，能短则短
- 以可用的软件来衡量项目进展
- 欢迎需求变化，利用这种变化来获得竞争优势
- 业务人员和开发人员在项目开发过程中应当共同工作
- 面对面的交流是最有效的沟通方式

敏捷开发原则-2

- 以有进取心的人为项目核心，充分支持和信任他们
- 只有不断关注技术和设计才能越来越敏捷
- 只有能自我管理的团队才能创造优秀的需求和设计
- 时时总结如何提高团队效率并付诸行动

什么是敏捷开发



一组软件开发思想的统称
以一组软件开发方法论为代表
具体体现为一系列相互支撑的概念、工具和最佳实践

敏捷开发的做法：以XP为例

如果.....

发挥到极致就变成.....

满足客户需求很重要

那就用客户的语言和行为来指导功能的开发
(Behavior Driven Development)

客户表达能力不强

那就请客户代表和团队人员一起工作

测试/单元测试能帮助提高质量

那就先写单元测试，从测试开始写程序（测试驱动的开发）

代码评审（Review）
可以找到错误

从一开始就处于“评审”状态：结对编程

计划没有变化快

那就别做详细的设计，和文档。通过增量开发，重构和频繁地发布来满足用户的需求

代码重构会提高质量

那就持续不断地重构

其他好方法.....

发挥到极限的做法.....

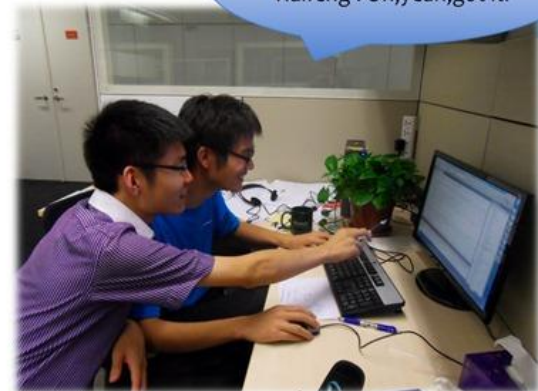
将最佳实践（Best Practice）发挥到极致（eXtreme）

敏捷实践

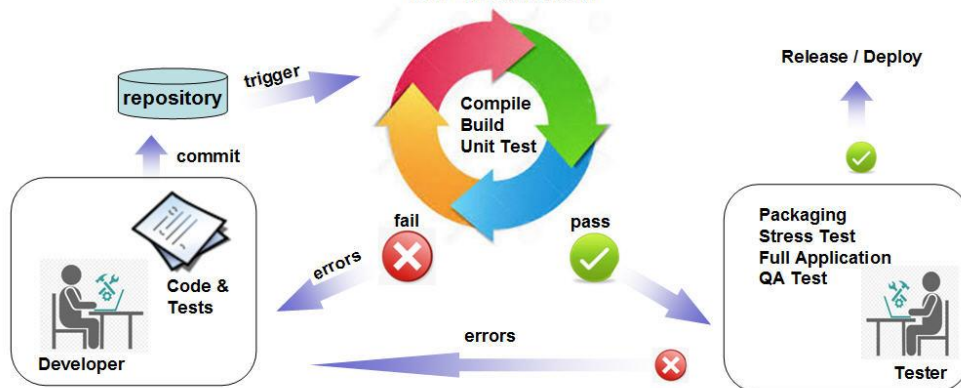


Pair Programming

Yueyi: Here is the point!
Haifeng: Oh,yeah,got it!



CONTINUOUS INTEGRATION



课堂讨论：敏捷开发

Class Discussion



课堂讨论：敏捷开发等同于杂乱无章、随心所欲的软件开发吗？

阅读建议

- 《软件工程》 第2、3章
- 《构建之法》 第5、6章

快速阅读后整理问题
在QQ群中提出并讨论

CS2001

软件工程

End

6. 软件过程与团队合作
— 软件过程与敏捷开发