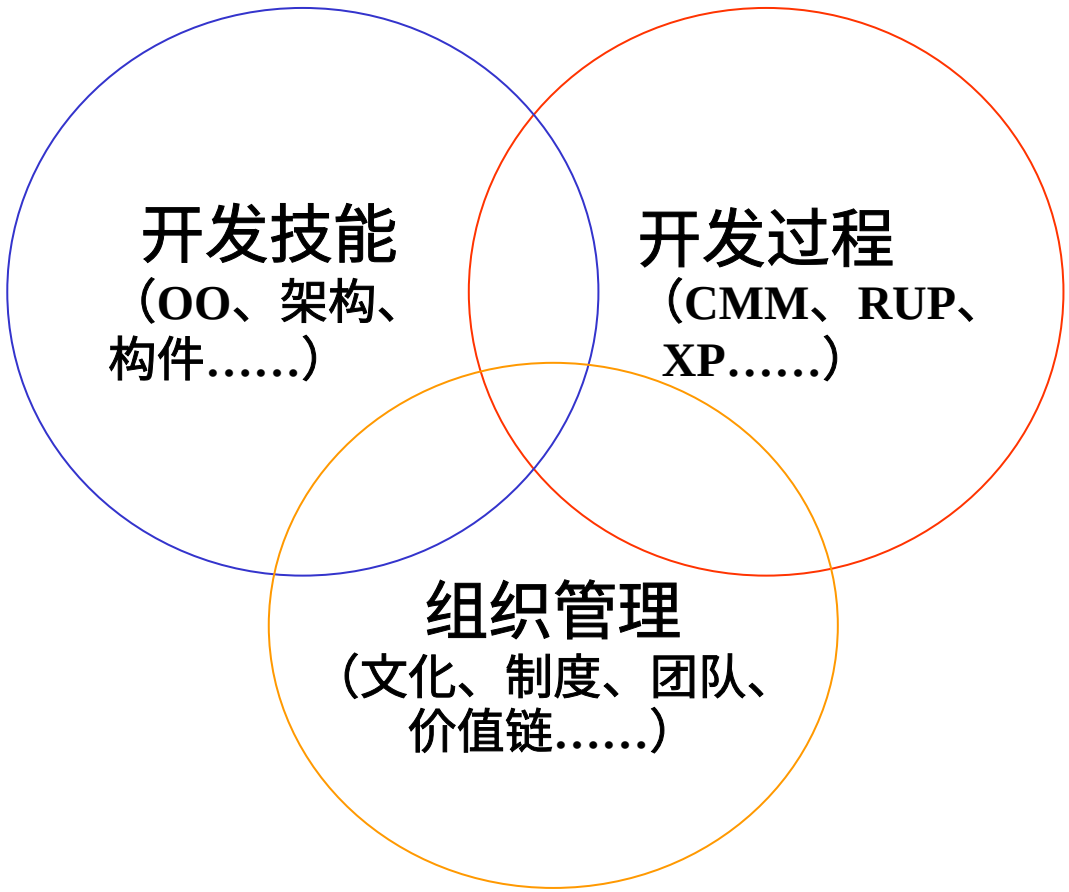
敏捷方法推进ＣＭＭ过程改进

大纲

1. **SPI、CMM、AP三者的关系**
2. 过程的多样性与选择
3. 敏捷方法综述
4. **AP与CMM/PSP/TSP的比较**
5. 中小企业**SPI**的策略和方法

软件企业的核心能力



国内软件企业问题

外部环境：

市场尚不规范健全，产业链资源分布不均

突出症状：

1)总体规模小，平均30人左右，87%以上100人以下。

2)竞争激烈，生存、壮大困难，因而普遍浮躁，无心练好内功。

3)软件技能、管理、过程的整体水平落后，营养失衡，发育不良。

国内软件企业的薄弱环节

1. 需求 – 定义模糊不清，适应不了快速变化
2. 架构 – 设计不重视，软件适应性、扩展性、可重用性差
3. 测试 – 忽视、轻视事先测试，事后弥补代价高
4. 评审– 往往走过场，发现、纠正问题的效用低
5. 过程 – 无规范、文档化的软件过程，或过程执行、管理的成熟度低，实用性差
6. 文化 – 凝聚力、协作精神缺失，团队不稳定
7. 客户– 与客户沟通不畅，市场压力大
8. 管理 – 技术管理落后，与生产实际脱节，效率低

……

软件过程改进

- 过程改进是一种系统化的、涉及组织变革的企业战略管理行为。
- ***Improvement on Demand***。SPI不应只有一种形式，企业应该根据自己的实际情况量力而行。
- SPI需要认真规划并持续改进，应该短期见效、长期获益。

SEI的IDEAL组织改进模型
(http://www.sei.cmu.edu/ideal/ideal.html)

SW- CMM

- SW-CMM是对过程进行诊断、衡量、比较的一个基准，提供了一个SPI的框架、阶梯和路线图。
- SW-CMM包括5个级别、18个关键过程域、52个目标、316个关键做法。

级别	关注重点	KPA
优化级	持续过程改进	DP
		TCM
		PCM
可管理级	产品和过程质量	QPM
		SQM
已定义级	工程过程与组织支撑	OPF
		OPD
		TP
		ISM
		SPE
		IC
		PR
可重复级	项目管理过程	RM
		SPP
		SPTO
		SSM
		SQA
		SCM
初始级	个人能力和英雄行为	

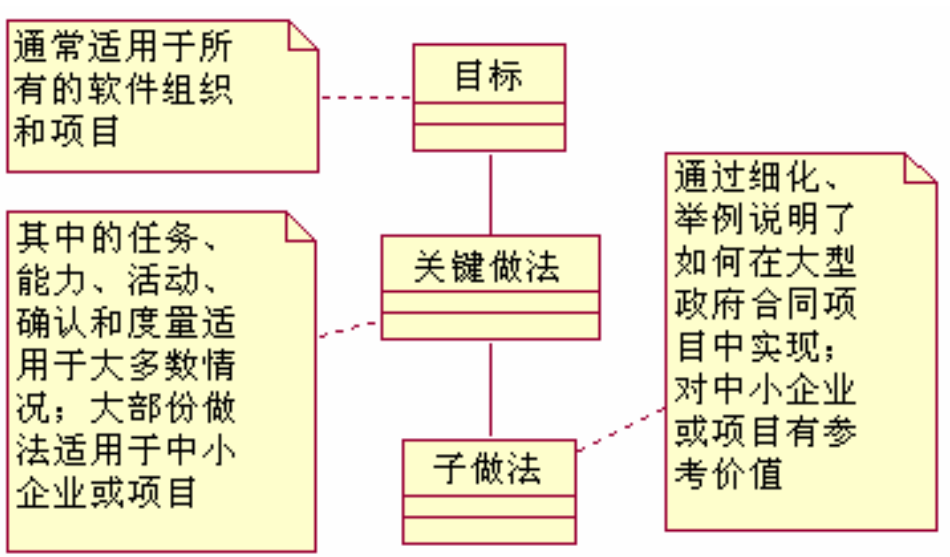
CMM特点

- CMM是一套适用面很广的通用过程实践标准，但CMM本身不是过程或方法论！
- CMM告诉组织为了系统化地建立、实施和改进软件过程应该做些什么，达到什么目标，但没有说明如何做、采用哪些具体技术、策略和方法。（PSP、TSP、RUP、XP……）
- CMM重视系统性、制度化和度量，强调提高过程的可靠性、可见性、可预测性和可管理性。
- 实施CMM要求组织在过程制度化建设上付出大量努力，通常被认为是重载的模型。

CMM的局限

“制订CMM的目的是向任何环境下的任何项目提供良好的软件工程和管理实践方法” —— **Mark Paulk**

CMM提出的背景是美国国防部希望以此来挑选、评价大型军工软件的承包商。



中小企业特点

- 产品不成熟、客户基础薄弱、无稳定收入，企业发展易受多方面不确定因素的影响。
- 生存壮大（而非更高的过程成熟度）是企业的首要目标！因而要求速度更快、成本更低、质量更好。
- 资源有限，组织不成熟，管理易变、反应式，因而采取的软件过程措施往往针对性强。
- 对过程灵活、实用的要求大于程式化、固定化。

CMM做法

某些不一定适用于中小企业的做法：

- **CMM 的定量控制要求工程师每天记录开发进度、工作量，并将相关数据填入报表，以便准确统计一个项目所需的工作量和产品规模。**
- **CMM实施周期很长，评估和工具的成本很高，隐性投入更大。**

……

中小企业CMM-SPI的难题

- 如何确定SPI的范围？
- 如何产生、制备大量必要的文档？
- 如何获得必要的资源和工具？
- 如何获得预先定义的、获批准的、完备的需求？
- 如何执行有效的评审？
- 如何开展培训，说服、教育员工？
- 如何独立验证过程的CMM符合度？
- 如何以较低成本通过正式的CMM评估（CBA-IPI、SCE）并获得投资回报？

……

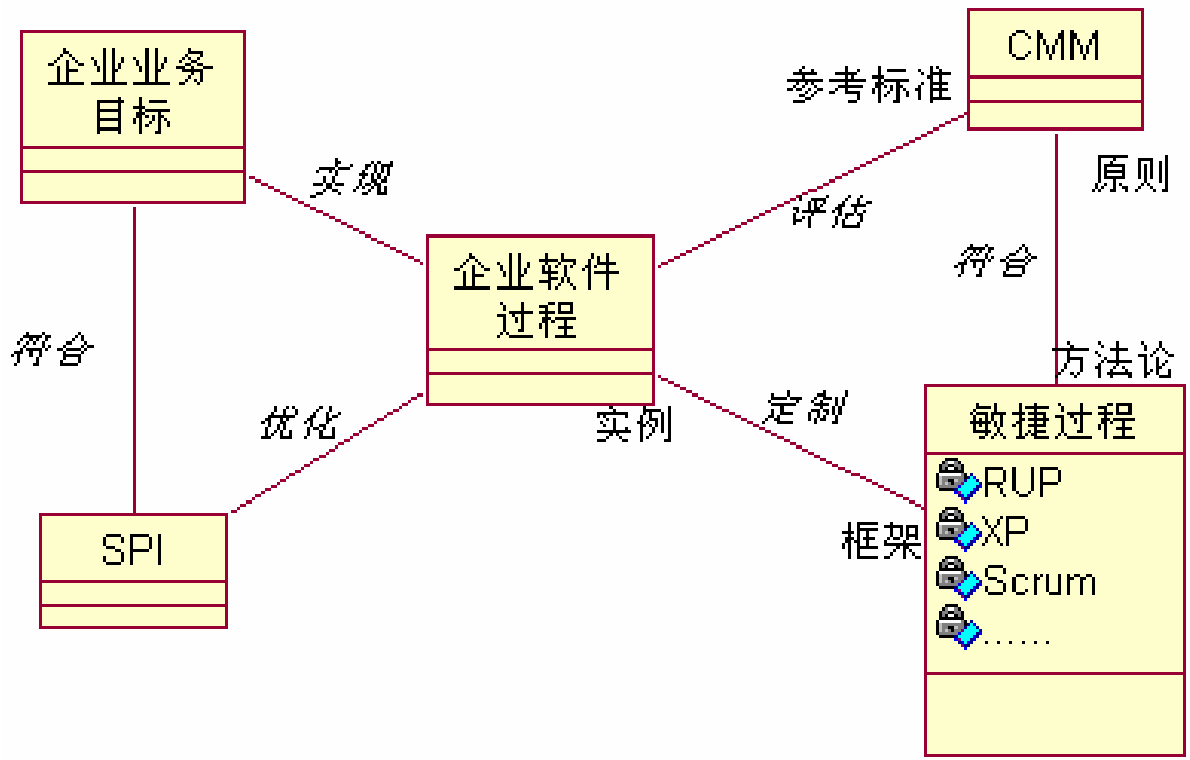
敏捷过程（AP，Agile Process）

“敏捷”的含义：

轻巧、机敏、迅捷、灵活、活力、高效……

- 轻载（过程、工件最小化）不等同于敏捷。
- 敏捷过程很容易适应变化并迅速做出自我调整，在保证质量的前提下，做到文档、度量适度（just enough），实现企业效益的最大化。
- 敏捷应该是所有成功组织（包括大中型企业）的目标！

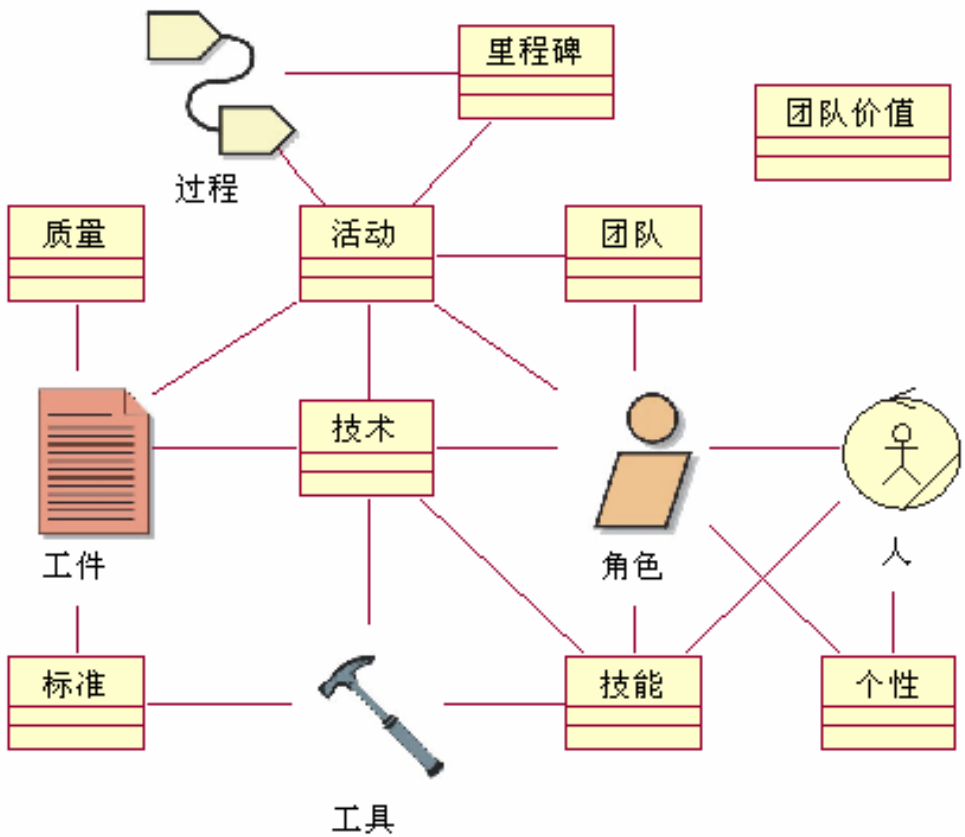
CMM、AP、SPI的关系



大纲

1. **CMM、SPI、AP三者的关系**
2. **过程的多样性与选择** ⟵
3. **敏捷方法综述**
4. **AP与CMM/PSP/TSP的比较**
5. **中小企业SPI的策略和方法**

过程组成



过程管理的内容

- 过程互动：

  信息流、工件流、
  控制、通讯、定时、依赖、并发……

- 过程管理任务：

  定义、实施、执行、
  记录、度量、评估、控制、预测、
  培训、学习、交流、
  理解、分析、比较、评价、改进、验证……

过程的多样性

“One Size does not Fit All”

- 具体环境：

  项目、产品（军用、民用等）、

  资源、团队、文化、地域（集中、分布等）……
- 层次：

  组织过程、项目过程、团队过程、个人过程
- 业务目标
- 开发类型：

  新产品、重用、COTS、维护、服务、产品线……

最佳过程？

- 一个项目的最佳过程是这个项目所能负担的最小过程。
- 过程的多样性决定了CMM以及SPI实施的多样化。
- 通过CMM评估的努力是复杂的、高成本的，但过程改进的有效性与复杂性、高成本之间没有必然联系。

过程多样化、敏捷化

据国际著名咨询机构Cutter Consortium对全球200位IS/IT经理所做的调查——

- 3个占优的重载方法：

  51% Rational Unified Process

  27% CMM

  26% ISO 9000

- 到2003年，大约50%的被调查者预计其50%以上的项目会使用敏捷方法；14%的被调查者认为其所有的项目会使用敏捷方法。

From THE DECISION IS IN: AGILE VERSUS HEAVY METHODOLOGIES, VOL. 2, NO. 19, by Robert Charette, Senior Consultant, Cutter Consortium

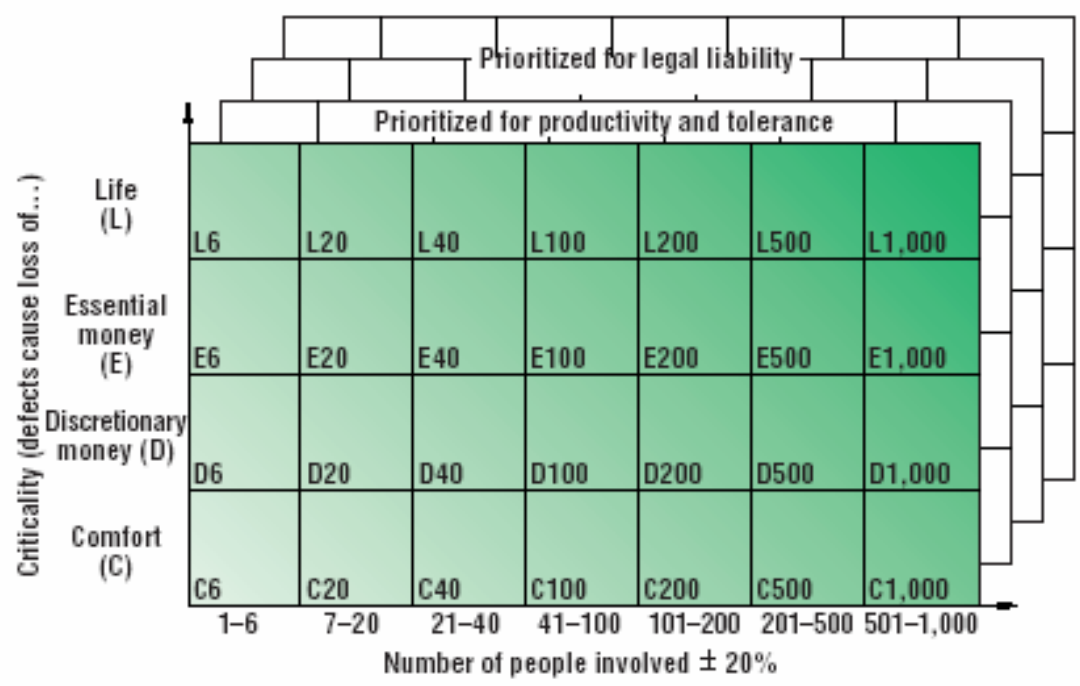
过程方法选择原则

1. 越大的团体需要越大的方法论。
2. 越关键的系统（未发现的缺陷将产生更严重的灾害）在构建的正确性方面需要越多的透明度（更大的密度）。
3. 在方法论大小或密度上相对小的增加会引起项目成本相对大的增长。
4. 最有效的沟通（交换意见）方式是面对面的互动，例如在白板前的讨论。

——Alistair Cockburn, *Selecting a Project's Methodology*

过程方法选择框架

“不同的项目需要不同的方法论”



——Alistair Cockburn, *Selecting a Project’s Methodology*

大纲

1. **CMM、SPI、AP三者的关系**
2. **过程的多样性与选择**
3. **敏捷方法综述** ⟵
4. **AP与CMM/PSP/TSP的比较**
5. **中小企业SPI的策略和方法**

敏捷联盟

- **2001年美国成立了AgileAlliance (www.agilealliance.com)，并发表了《敏捷软件开发宣言》。**
- **敏捷价值观：**

“注重个人及互动胜于过程和工具”

“注重可用的软件胜于详尽的文档”

“注重客户协作胜于合同谈判”

“注重响应变化胜于恪守计划”

张恂译自www.agilemanifesto.org

敏捷方法

- *Crystal*
- *Agile Software Development*
- *Feature Driven Development*
- *Scrum*
- *Rational Unified Process*
- *dx*
- *eXtreme Programming*
- *DSDM*
- *Agile Modeling*
- *Agile Project Management*

Crystal

- 由Alistair Cockburn提出(*The Humans and Technology Manifesto of Software Development*)，2001年2月宣布将于ASD融合。
- Crystal方法族：

  Clear, Yellow, Orange, Red, Maroon, Blue,Violet ...
- 强调人是第1位的要素。
- 探讨了纪律要求最不严格却依然可行的方法论，为了使过程易于执行而有意识地损失一些生产率。

crystalmethodologies.org

ASD

- 由Jim Highsmith根据复杂自适应系统（混沌）理论提出。
- 软件开发包括3个非线性、重叠的阶段：

speculation（思考）
collaboration（协作）
learning（学习）

FDD

由Jeff de Luca、Peter Codd（TogetherSoft）提出。

过程阶段：

1. 开发总体模型
2. 建立特性列表
3. 特性计划
4. 特性设计
5. 特性构造

ETVX：

1. 入口标准
2. 任务
3. 确认
4. 出口标准

角色：

1. 领域成员
2. 总架构师
3. 主程序员
4. 类负责人
5. 特性小组
6. 发布经理

R Palmer, John M. Felsing, *A Practical Guide to Feature-Driven Development by Stephen,* Prentice Hall PTR, February 11, 2002

Scrum

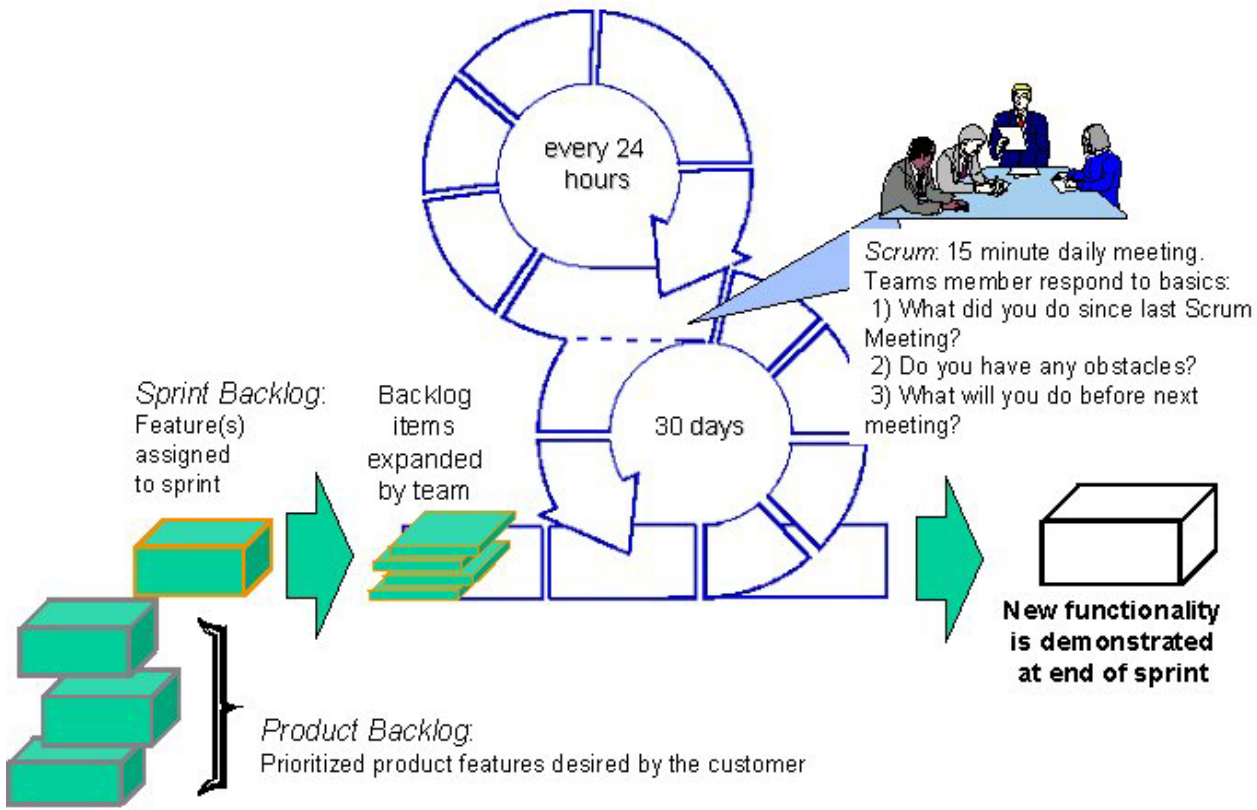
- 1993年Jeff Sutherland在其Easel团队中首次采用（后Ken Schwaber、Mike Beedle进行了改进）。
- 组成：

  争球会议（Scrum Meeting）、积压任务表（ Backlog）、争球队长（Scrum Master）、短跑（ Sprint ）.....
- Scrum的管理和控制机制实际上适用于任何类型的项目。
- 应用实例：

  著名的AG Communication Systems公司在多个项目中成功运用了Scrum以解决通信软件开发的难题 （Linda Rising）

www.jeffsutherland.org/scrum/

Scrum模型



from www.controlchaos.com

极限编程（XP）

- 由Kent Beck、Ward Cunningham、Ron Jeffries等人提出反响最大、最为完善的敏捷过程方法。
- 应用实例:

  1996-1999年DaimlerChrysle C3薪资管理系统
- 价值观：

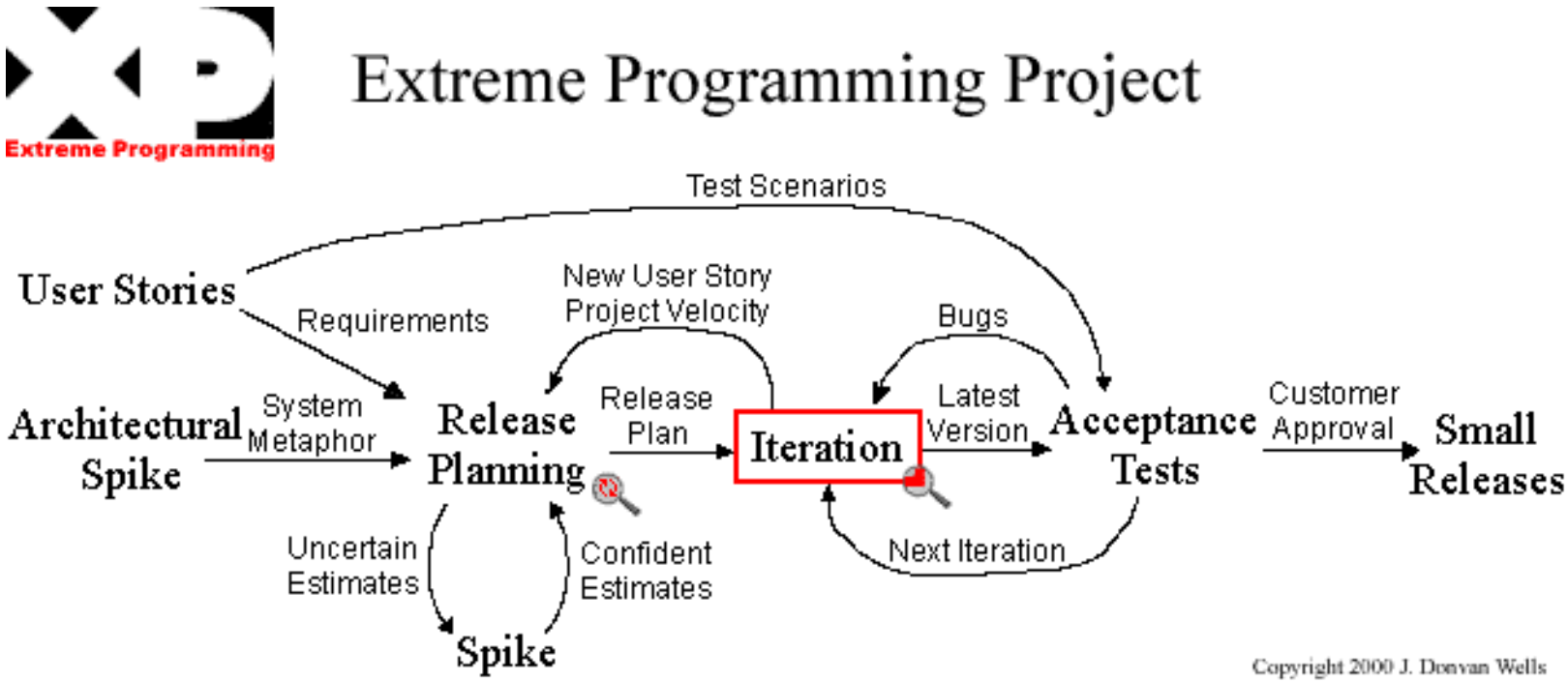
  沟通、反馈、简化、勇气
- 特点：

  测试成为开发的核心;

  纪律性与灵活性巧妙结合.

www.extremeprogramming.org

XP项目周期


Extreme Programming
Extreme Programming Project
Test Scenarios
User Stories
Requirements
New User Story
Project Velocity
Bugs
Architectural
Spike
System
Metaphor
Release
Planning
Release
Plan
Iteration
Latest
Version
Acceptance
Tests
Customer
Approval
Small
Releases
Uncertain
Estimates
Confident
Estimates
Spike
Next Iteration
Copyright 2000 J. Donvan Wells

XP关键做法

1. 现场客户（On-site Customer）
2. 计划博弈（Planning Game）
3. 系统隐喻（System Metaphor）
4. 简化设计（Simple Design）
5. 集体拥有代码（Collective Code Ownership）
6. 结对编程（Pair Programming）
7. 测试驱动（Test-driven）
8. 小型发布（Small Releases）
9. 重构（Refactoring）
10. 持续集成（Continuous integration）
11. 每周40小时工作制（40-hour Weeks）
12. 代码规范（Coding Standards）

计划博弈

- XP要求结合业务和技术情况，快速确定下一次发布的范围。在项目计划的4要素（费用、时间、质量和范围）中，由客户选择3个，而程序员可以选择剩下的1个。

- 通常客户从业务角度确定项目范围、需求优先级和开发进度，开发人员则做出具体的成本和技术估计。

- XP强调简短和突发性的计划，有时只用几个小时甚至几分钟就能完成，而且可以随时按需进行多次计划。

系统隐喻

- XP通过一个简单的关于整个系统如何运作的隐喻性描述（story）来指导全部开发。
- 隐喻可以看作是一种高层次的系统构想，通常包含了一些可以参照和比较的类和模式，它还给出了后续开发所使用的命名规则。
- XP不需要事先进行详细地架构设计。

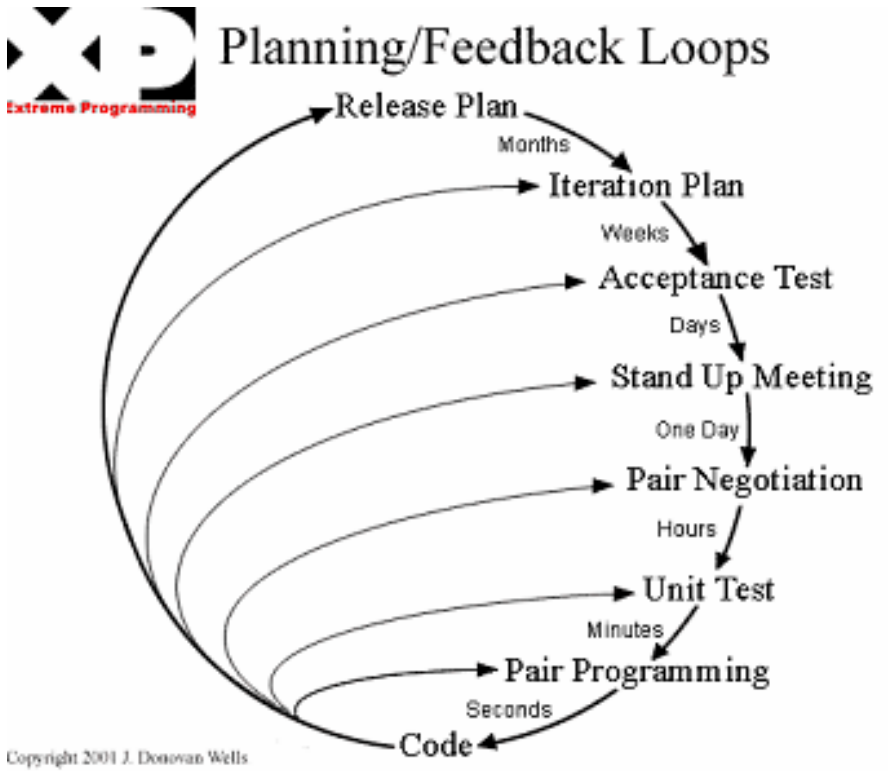
结对编程

- 由两名程序员在同一台电脑上结成对子共同编写解决同一问题的代码。
- 通常一个人写代码，另一个人同时负责保证代码的正确性和可读性，比如编写单元测试程序、进行代码走查。
- PP可以看作是一种非正式的持续进行的同行评审（peer review）。

重构

- 重构是指在不改变系统行为的前提下，重新调整、优化系统的内部结构以减少复杂性、消除冗余、增加灵活性和提高性能。

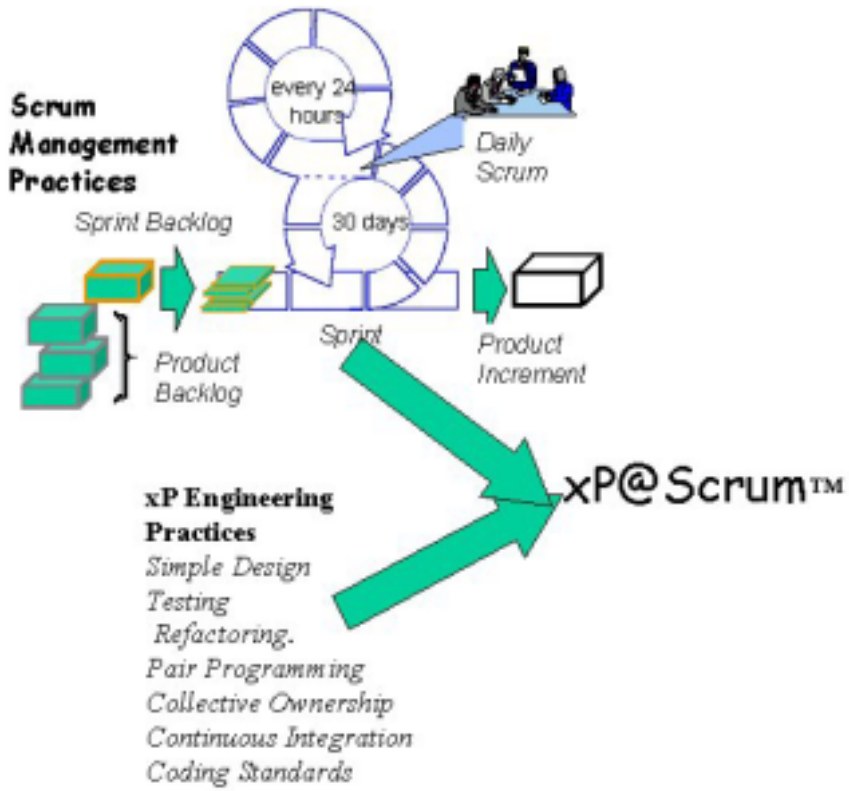
测试驱动



“先写测试，后编码”

XP 与Scrum集成

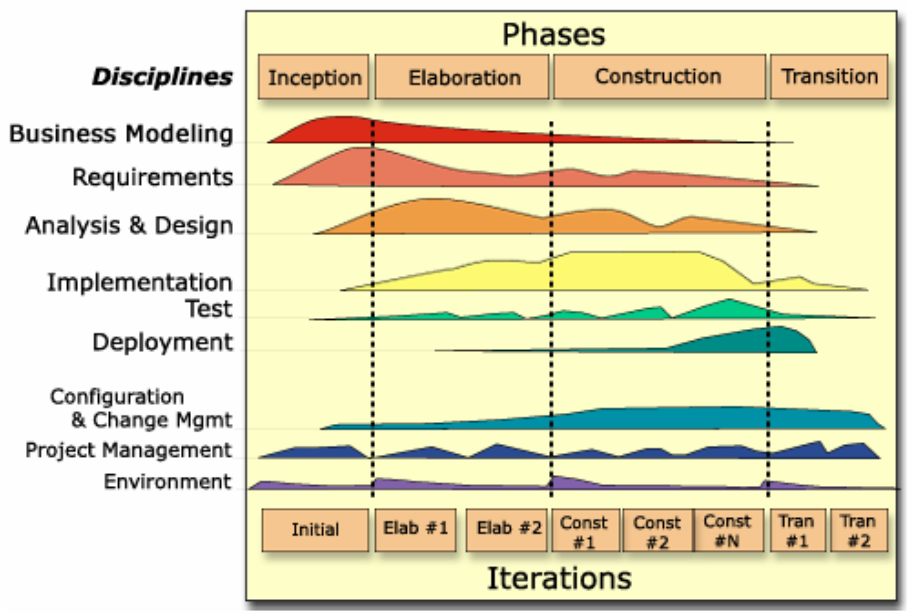
- **Scrum提供了敏捷的管理机制。**
- **XP提供了敏捷的集成工程方法。**



from www.controlchaos.com

Rational Unified Process

RUP是一个风险驱动的、基于UML和构件式架构的迭代、递增型开发过程框架（重载或轻载）。



→ 4个阶段
→ 9个科目
→ 57种活动
→ 270个活动步骤
→ 114种工件
→ 38种角色

RUP最佳实践方法

迭代式开发

需求管理

构件式架构

基于UML的可视化建模

持续校验质量

变更管理

RUP与XP的共性

- 基础都是面向对象方法（取代传统的结构化方法）
- 都重视代码、文档的最小化和设计的简化
- 采用动态适应变化的演进式迭代周期（取代传统的瀑布型生命周期）
- 强调需求和测试管理
- 鼓励用户积极参与

RUP与XP的区别

- XP以代码为中心，编码和设计活动融为一体，弱化了架构的概念。
- RUP过程通常以架构为中心，细化阶段的主要目的就是构造出一个可运行的架构原型，作为将来添加需求功能的稳固基础。
- XP不包含业务建模、部署、过程管理等概念。
- RUP适合各种规模的项目，XP只适用于小团队。

有趣的dx过程

- 由Robert Martin（Object Mentor）提出。
- 在RUP的框架下，定制出一个非常接近XP的过程，展现了RUP的灵活性，说明其过程不一定是重载的。
- 特点：

  用例驱动（非Story）

  细化和构造阶段没有明显区分（先开发架构）

《敏捷宣言》12条原则

1. 最优先的目标是通过尽早地、持续地交付有价值的软件来满足客户。
2. 欢迎需求变化，甚至在开发后期。敏捷过程控制、利用变化帮助客户取得竞争优势。
3. 频繁交付可用的软件，间隔从两周到两个月，偏爱更短的时间尺度。
4. 在整个项目中业务人员和开发人员必须每天在一起工作。
5. 以积极主动的员工为核心建立项目，给予他们所需的环境和支持，信任他们能够完成工作。
6. 在开发团队内外传递信息最有效率和效果的方法是面对面的交流。

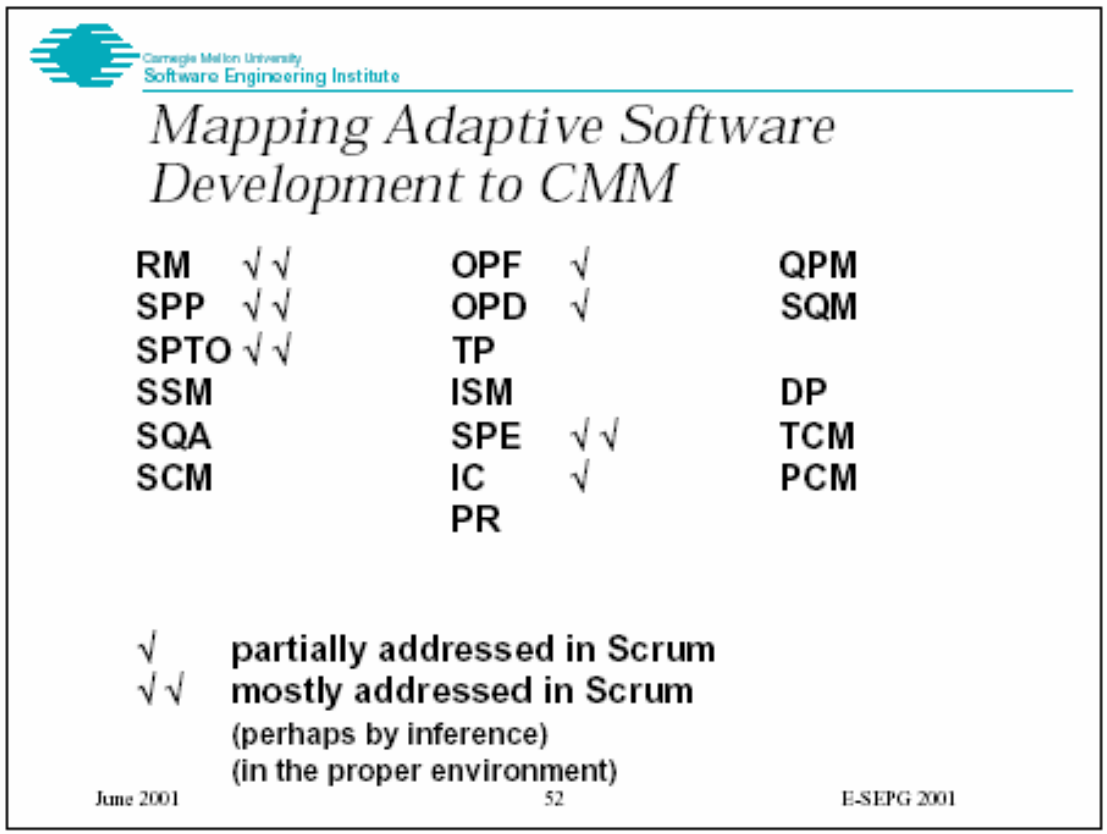
《敏捷宣言》12条原则

7. 可用的软件是进展的主要度量指标。
8. 敏捷过程提倡可持续发展。发起人、开发者和用户应始终保持稳定的步调。
9. 简化——使必要的工作最小化的艺术——是关键。
10. 持续关注技术上的精益求精和良好的设计以增强敏捷性。
11. 最好的架构、需求和设计产生于自我组织的团队。
12. 团队定期地对运作如何更加有效进行反思，并相应地调整、校正自己的行为。

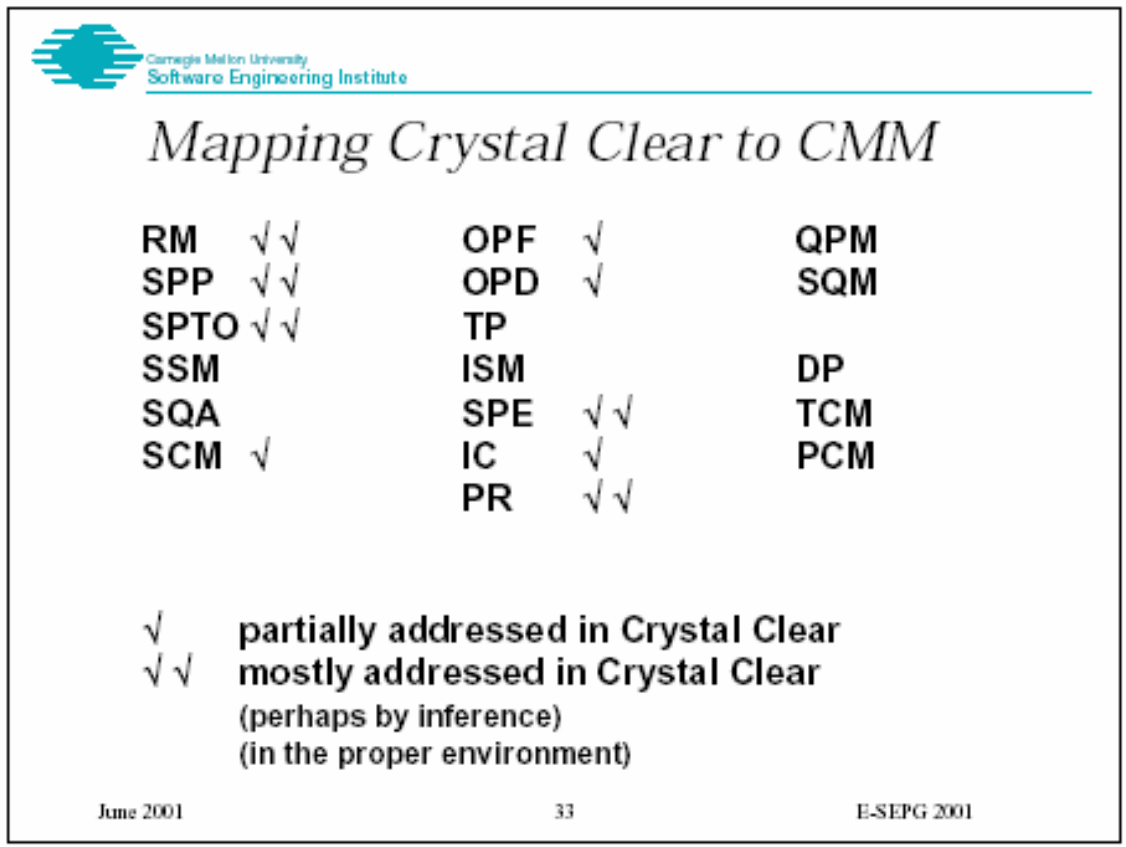
大纲

1. **CMM、SPI、AP三者的关系**
2. **过程的多样性与选择**
3. **敏捷方法综述**
4. **AP与CMM/PSP/TSP的比较** ⟵
5. **中小企业SPI的策略和方法**

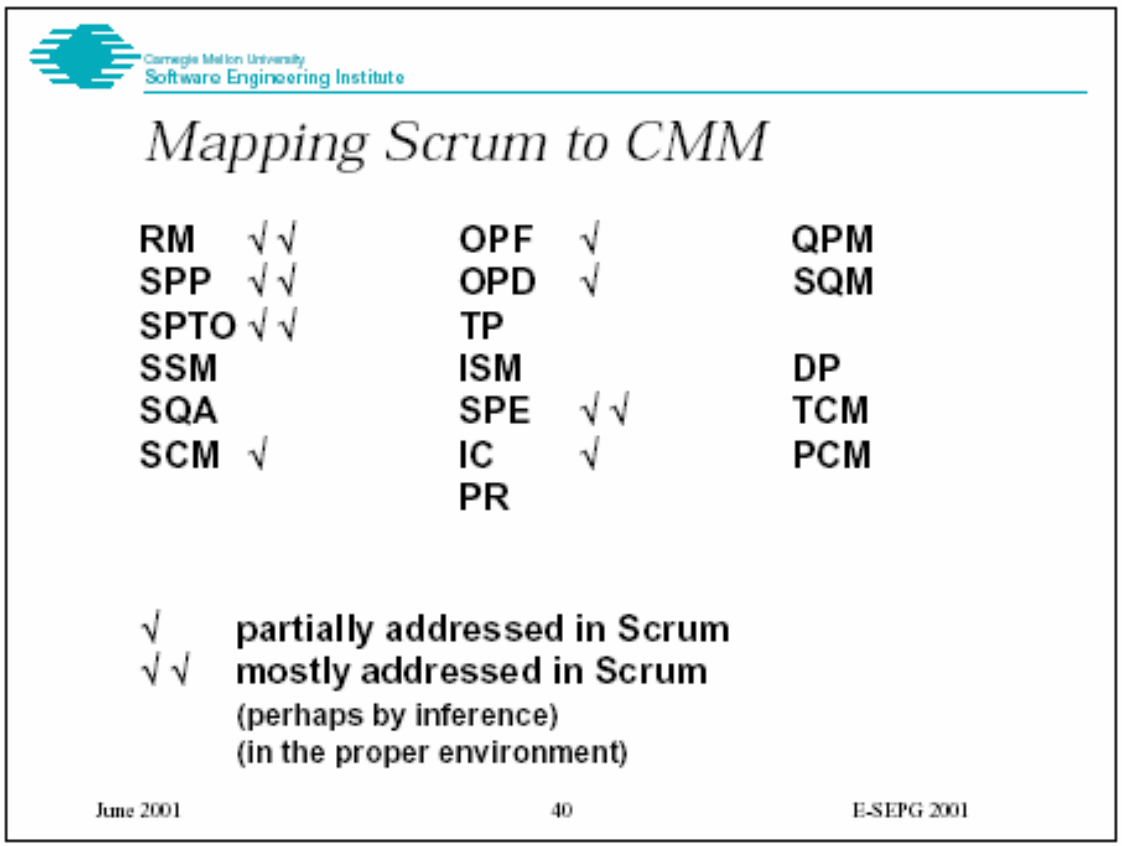
ASD与CMM KPA


Carnegie Mellon University
Software Engineering Institute
Mapping Adaptive Software Development to CMM
RM √ √
SPP √ √
SPTO √ √
SSM
SQA
SCM
OPF √
OPD √
TP
ISM
SPE √ √
IC √
PR
QPM
SQM
DP
TCM
PCM
√ partially addressed in Scrum
√ √ mostly addressed in Scrum
(perhaps by inference)
(in the proper environment)
June 2001
52
E-SEPG 2001

Crystal Clear与CMM KPA



Scrum与CMM KPA



XP与CMM KPA

Carnegie Mellon University
Software Engineering Institute

Mapping XP to CMM

RM	√ √	OPF	√	QPM	
SPP	√ √	OPD	√	SQM	
SPTO	√ √	TP			
SSM		ISM		DP	√
SQA	√	SPE	√	TCM	
SCM	√ √	IC	√ √	PCM	
		PR	√		

√ partially addressed in XP

√ √ mostly addressed in XP

(perhaps by inference)

(in the appropriate environment)

XP与CMM制度保障

KPA 公共特性	措施	满足度
执行承诺	方针政策	−−
	领导与支持	−−
执行能力	组织结构	+
	资源和经费	+
	培训	+
度量与分析	度量	+
实施确认	高层管理监控	−−
	项目管理监控	++
	软件质量保证	+

——Mark Paulk, *Extreme Programming from a CMM Perspective*

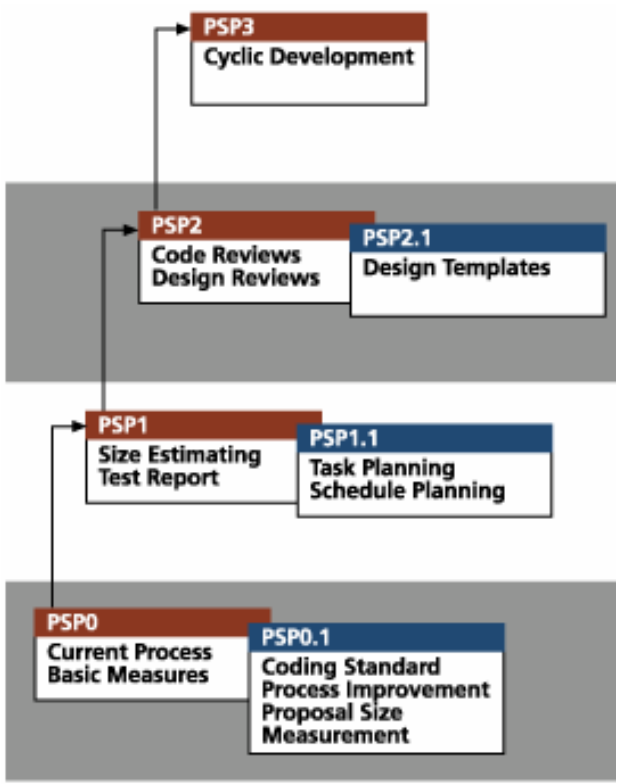
AP与CMM的比较

- AP的做法基本符合CMM目标和KPA，满足了CMM L2-L3大部分KPA的要求，但基本没有涉及CMM L4-L5。
- AP结合了具体的开发技术和技能，可操作性好，CMM则更关注过程的组织管理原则与目标。
- AP缺少“制度化”措施，即使良好的工程和管理实践制度化的关键基础设施和管理要件。

PSP

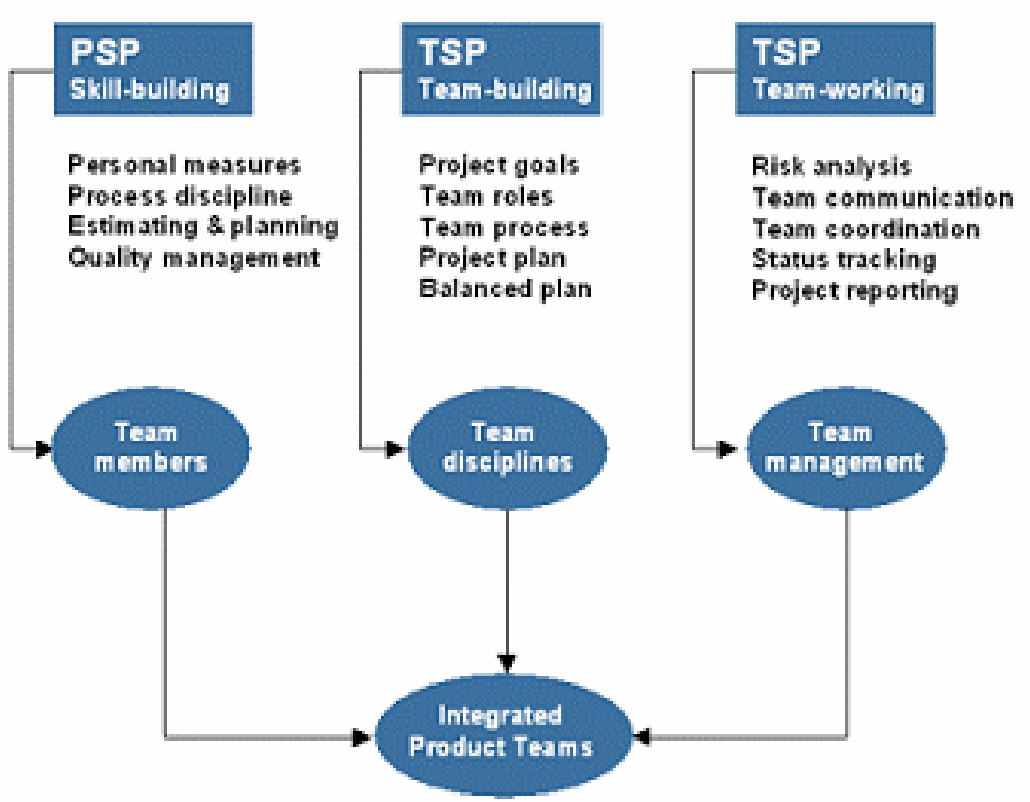
PSP是一种可用于控制、管理和改进个人工作方式的自我持续改进过程，是一个包括软件开发表格、指南和规程的结构化框架。

PSP以统计过程控制为基础，是一种定量软件过程方法。



TSP

TSP在PSP的基础上，通过度量数据的定义、收集、分析、核查和评估来提高对进度、工作量的估计能力和计划的精确度，减少代码缺陷，以减少测试时间与成本，提高总体生产率。



TSP特点

- 适合规模为3～20人开发大型高可靠性软件的小组。
- 帮助建立适合自我管理的团队，能够有效地规划和跟踪、掌握过程和计划。
- 通过使CMM L5的做法成为常规、期望的行为加速过程改进。
- 告诉管理者如何训练、激励团队以保持最高的效能。
- 向高成熟度的组织提供改进指导。

PSP/TSP与CMM KPA

	PSP	TSP		PSP	TSP		PSP	TSP
RM		**X**	**OPF**	**X**	**X**	**QPM**	**X**	**X**
SPP	**X**	**X**	**OPD**	**X**	**X**	**SQM**	**X**	**X**
SPTO	**X**	**X**	**TP**					
SQA		**X**	**ISM**	**X**	**X**	**DP**	**X**	**X**
SCM		**X**	**SPE**	**X**	**X**	**TCM**	**X**	**X**
SSM			**IC**		**X**	**PCM**	**X**	**X**
			PR	**X**	**X**			

Watts Humphrey, *Pathways to Process Maturity:The Personal Software Process and Team Software Process*

AP与PSP/TSP的目标

提高产品质量

控制成本和进度

缩短产品周期

改进过程

提高生产率

AP与PSP/TSP的区别

1. AP与PSP/TSP的适用环境和对象不同。
2. PSP/TSP强调对过程、产品质量全面精确的度量，前期投入大；AP的焦点在于可用产品的持续性开发，并使度量最小化。
3. PSP/TSP依赖于大量的数据、表格和文档；AP注重人与人之间的直接沟通，尽量减少工件。
4. PSP/TSP强调程式化、固定的流程；AP强调柔性和应变能力。
5. PSP/TSP的培训、实施成本很高。

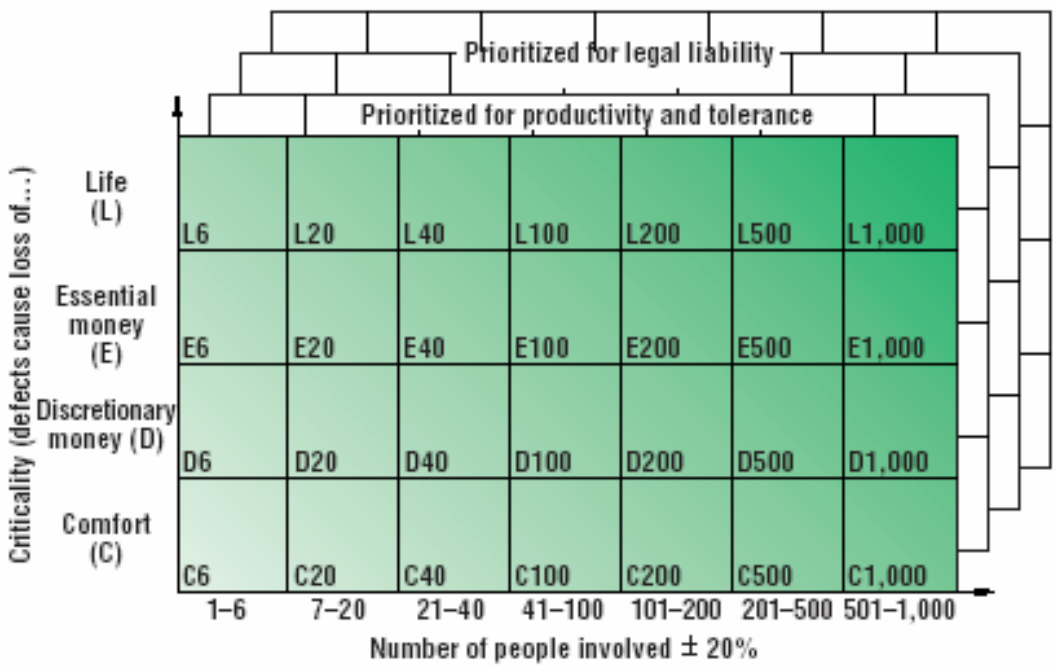
度量的意义

检验任何过程的好坏应该以交付产品的质量或项目的ROI为最终指标，不应片面地最求过程度量的完备性和成熟度。

- 如果没有工具的有力支持，PSP/TSP数据收集、维护成本很高，度量的准确性、实效性也很难保证。
- 在中小企业的商业软件产品、工程开发中，及时推出满足客户需求、刚好适用的产品和服务是第一位的，适用AP（*just enough measurement*）。
- PSP/TSP的基调是质量优先于效率和成本，其对过程的管理和估计是基于对历史统计数据的收集分析，需要大量经验数据的长期积累，适用于环境和需求相对稳定、综合实力强、开发质量关键软件的大中型组织和需要对过程能力做预测、评价的大型工程、外包项目。

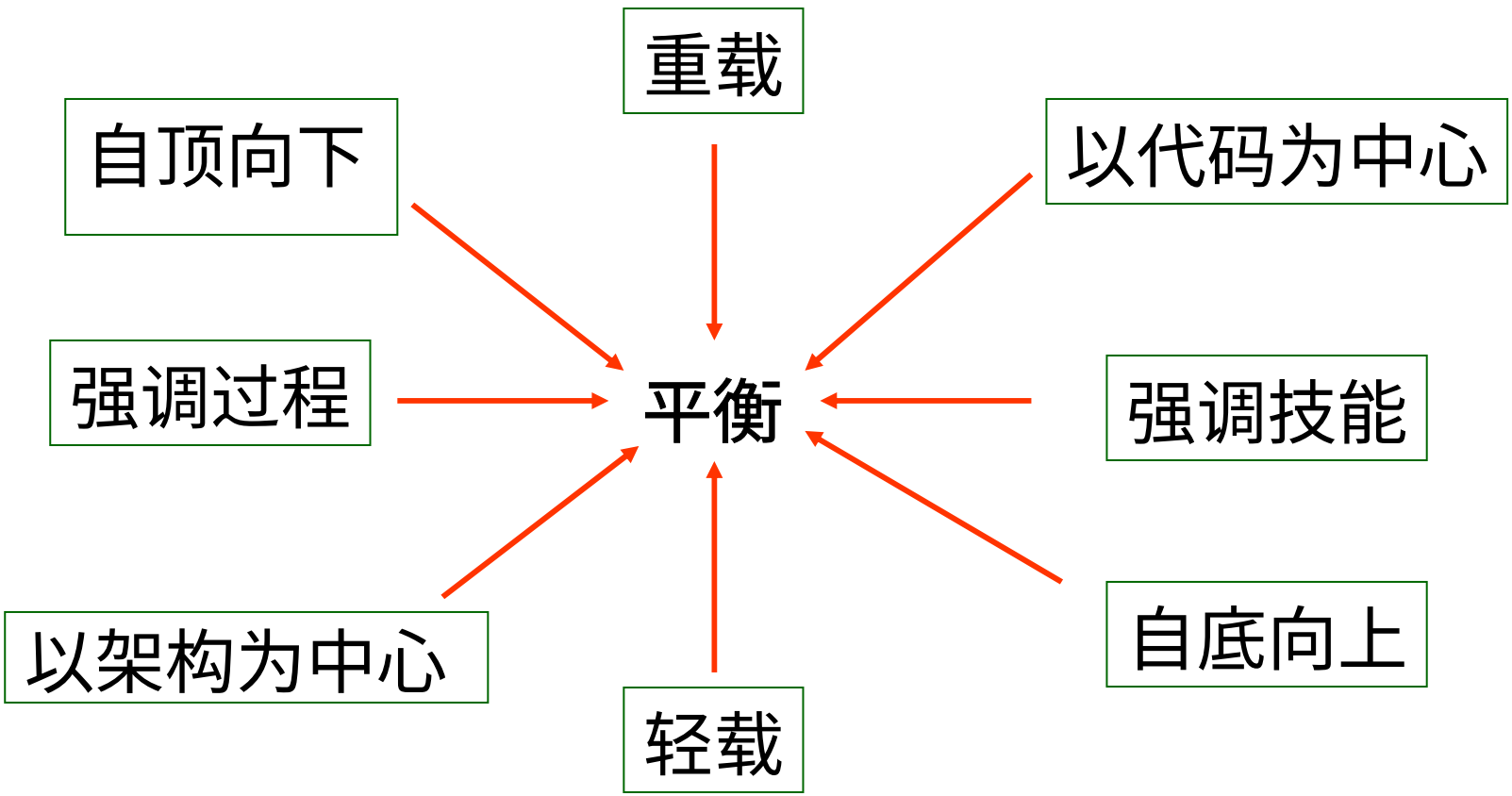
过程方法选择框架

“不同的项目需要不同的方法论”

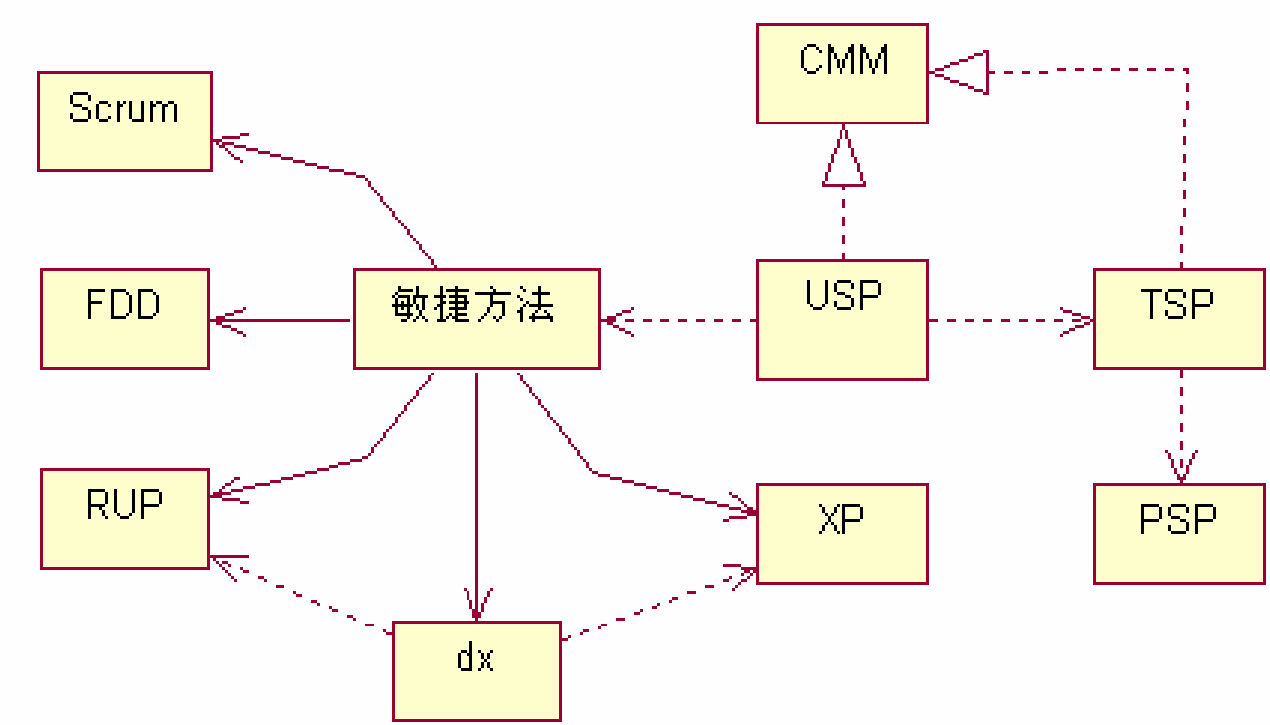


——Alistair Cockburn, *Selecting a Project’s Methodology*

成功之道——掌握平衡



统一软件过程（USP）



USP的特点

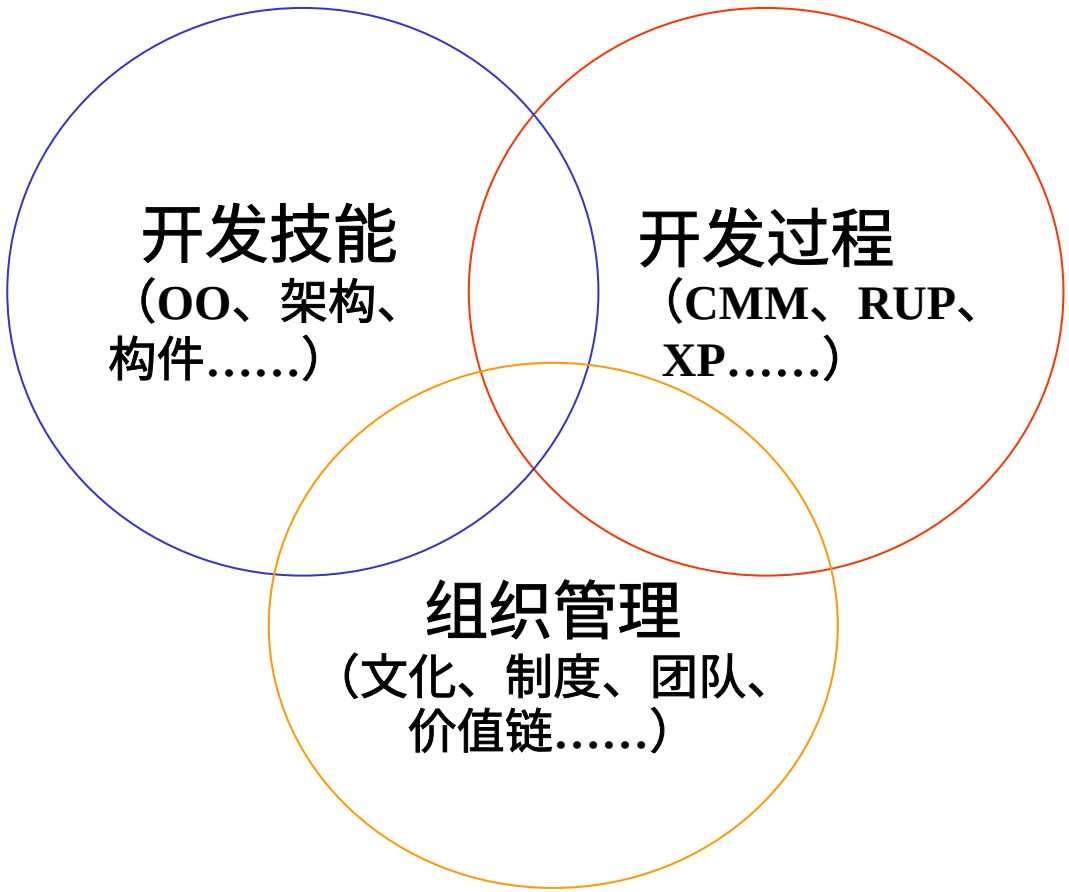
- USP是一套以敏捷思想为基础建立的、符合CMM L3要求的、统一的组织过程体系。
- USP融合了各种方法论的长处：

  RUP（架构、业务分析）、XP（测试、重构）、
  PSP/TSP（度量）、Scrum（团队管理）.....
- USP过程不是固定不变的，而是根据项目的实际情况自觉地对生命周期、具体做法和措施进行动态组合、调整、适应和优化。
- *Process on Demand*。在项目的不同阶段、不同团队、不同模块可分别采用不同的过程和方法。

大纲

1. **CMM、SPI、AP三者的关系**
2. **过程的多样性与选择**
3. **敏捷方法综述**
4. **AP与CMM/PSP/TSP的比较**
5. **中小企业SPI的策略和方法** ⟵

中小企业SPI策略



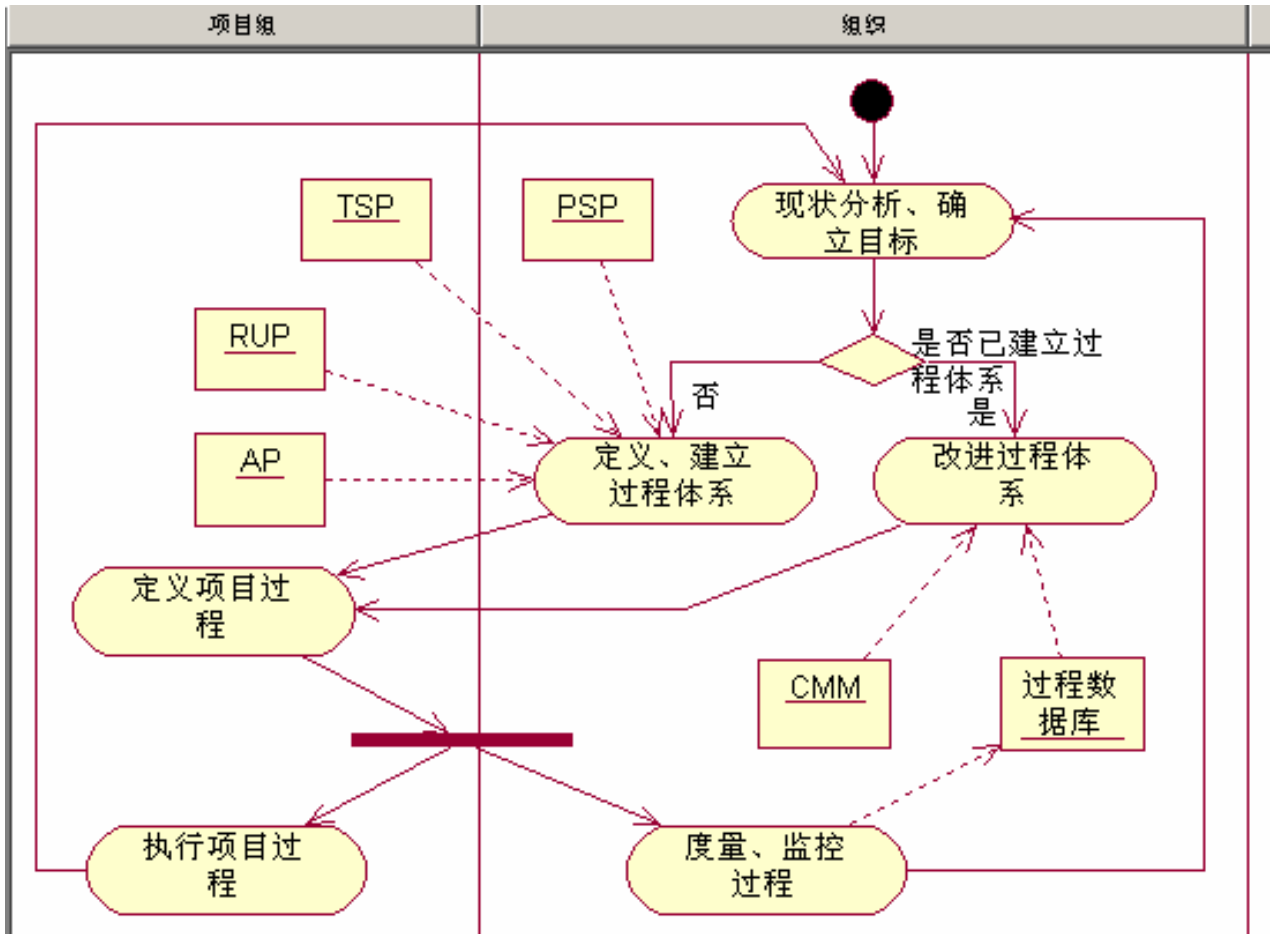
敏捷过程的意义

- 软件业的“敏捷过程”可与制造业的敏捷制造、柔性制造类比。
- 敏捷过程本着实用经验主义的思想，在资源条件有限的情况下，对质量、速度、成本作出巧妙有效的权衡，是中小企业实施SPI、达到CMM L3同类水平的高效、低成本的敏捷之路。
- 敏捷方法也提醒各类企业要以人为本、以客户为中心，时刻注意简化，不断提高效率，避免繁文缛节、官僚主义和形式主义。

对中小企业SPI的建议

- 从解决企业迫切问题入手，采取演进方式，巧妙地平衡短期利益和长期利益，紧密结合业务目标实施SPI。
- 以实施敏捷过程为起点，掌握CMM、PSP/TSP、XP、RUP、Scrum等方法的精髓，取长补短，定义并实施适合自身的、符合CMM要求的过程体系，逐步建立起敏捷、高效的企业组织和文化，获得良好的投资回报。
- 适当借助外力（诊断、文档、培训、咨询）以减少成本、少走弯路、加快发展。
- 倾听、重视开发人员和客户的抱怨。

中小企业SPI循环


项目组
组织
TSP
PSP
现状分析、确立目标
RUP
是否已建立过程体系
否
是
AP
定义、建立过程体系
改进过程体系
定义项目过程
CMM
过程数据库
执行项目过程
度量、监控过程

引入敏捷过程的方法

由弱到强：

1）在保持组织原有的开发过程和生命周期模型的情况下，借鉴、采取个别对项目有效的敏捷方法。

2）采用敏捷过程的生命周期模型和大部分做法，根据实际情况对个别做法进行调整或舍弃。

3）完全按照新的生命周期模型全面、严格地执行敏捷过程。

VRAPS模型

软件架构（Architecture）的5项组织原则：

- 构想（Vision）
- 节奏（Rhythm）
- 预见（Anticipation）
- 协作（Partnering）
- 简化（Simplification）

对SPI也适用？

《软件架构——组织原则与模式》，张恂译

结论

- 达到SPI的目的有多种途径，CMM/PSP/TSP不是教条和唯一选择。关键是如何提升企业的总体竞争能力，不可偏废某一方面。
- 实施敏捷过程是中国软件企业尤其中小企业持续过程改进的最有效途径之一。
- SPI的成功自始至终离不开管理者的职业判断和常识理解（Common Sense），不能人云亦云、随波逐流！

*ASTI Shanghai*的作用

- 客户化的专业系列培训
- 完善的咨询服务：
  - 诊断、分析、建议、解答
  - 项目教练、指导
  - 中小企业的编外“SEPG”
  - 软件外包
  - “ITS之源”（www.iturls.com）

  ……

展望

在CMM的框架下统一（混合）过程如何发展？

如何在大中型软件企业和项目中以及更高的CMM级别上引入敏捷方法？

敏捷工程、敏捷管理、敏捷组织、敏捷文化

……

研讨问题

1. 上海及华东地区软件企业SPI、CMM实施的现状与发展？
2. 敏捷过程、统一过程在CMM过程改进中有何作用，它们的关系如何？
3. 在软件企业日常管理、生产、SPI中有哪些难题、陷阱？
4. 有哪些好的SPI具体策略、办法和措施？
5. 如何实施敏捷过程？

……

参考资源

1. Martin Fowler, *the New Methodology*, http://martinfowler.com
2. Mark C. Paulk, *Using the Software CMM in Small Organizations*
3. Mark C. Paulk, *"Internet-Speed Processes" From a CMM Perspective*, E-SEPG 2001/6
4. 居德华,《*建设一个成熟的软件产业*》, http://www.iturls.com
5. 钱乐秋、张敬周、朱三元，《*Agile方法研究综述*》,http://www.iturls.com
6. 曲俊生,《*从一个项目谈XP在国内的应用*》, http://www-900.ibm.com/developerWorks/cn/java/l-xp/index.shtml?n-j-08021
7. 沈备军,《*敏捷软件过程的研究*》, http://www.iturls.com
8. 张恂,《*XP的价值和局限*》，http://www.iturls.com