

敏捷配置管理实施

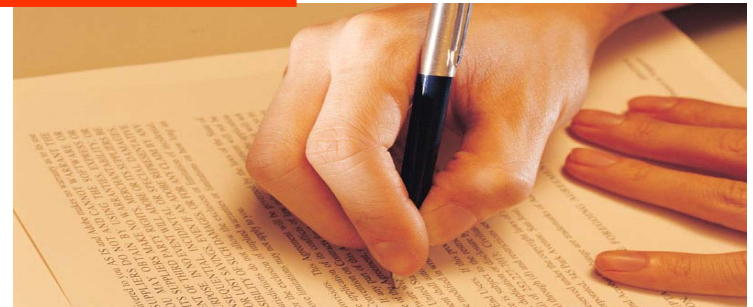


周小辉

软件工程大会

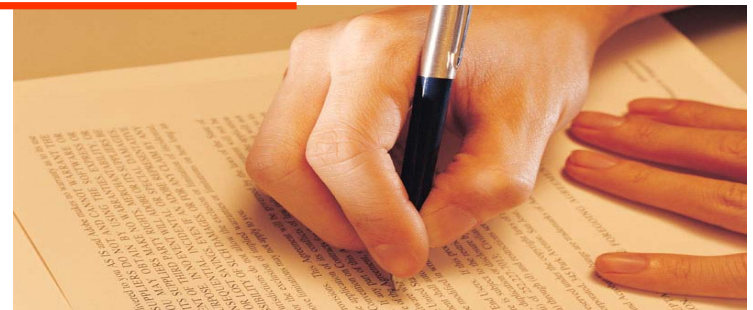
主要内容

- 1. 软件配置管理业界现状**
- 2. 配置管理实施规划**
- 3. 配置管理概述**
- 4. 融入敏捷思想**



1.软件配置管理业界现状

- 📁 SCM在软件开发过程中的地位日显突出
- 📁 厂商动作
- 📁 国内SCM应用现状存在很大改进空间

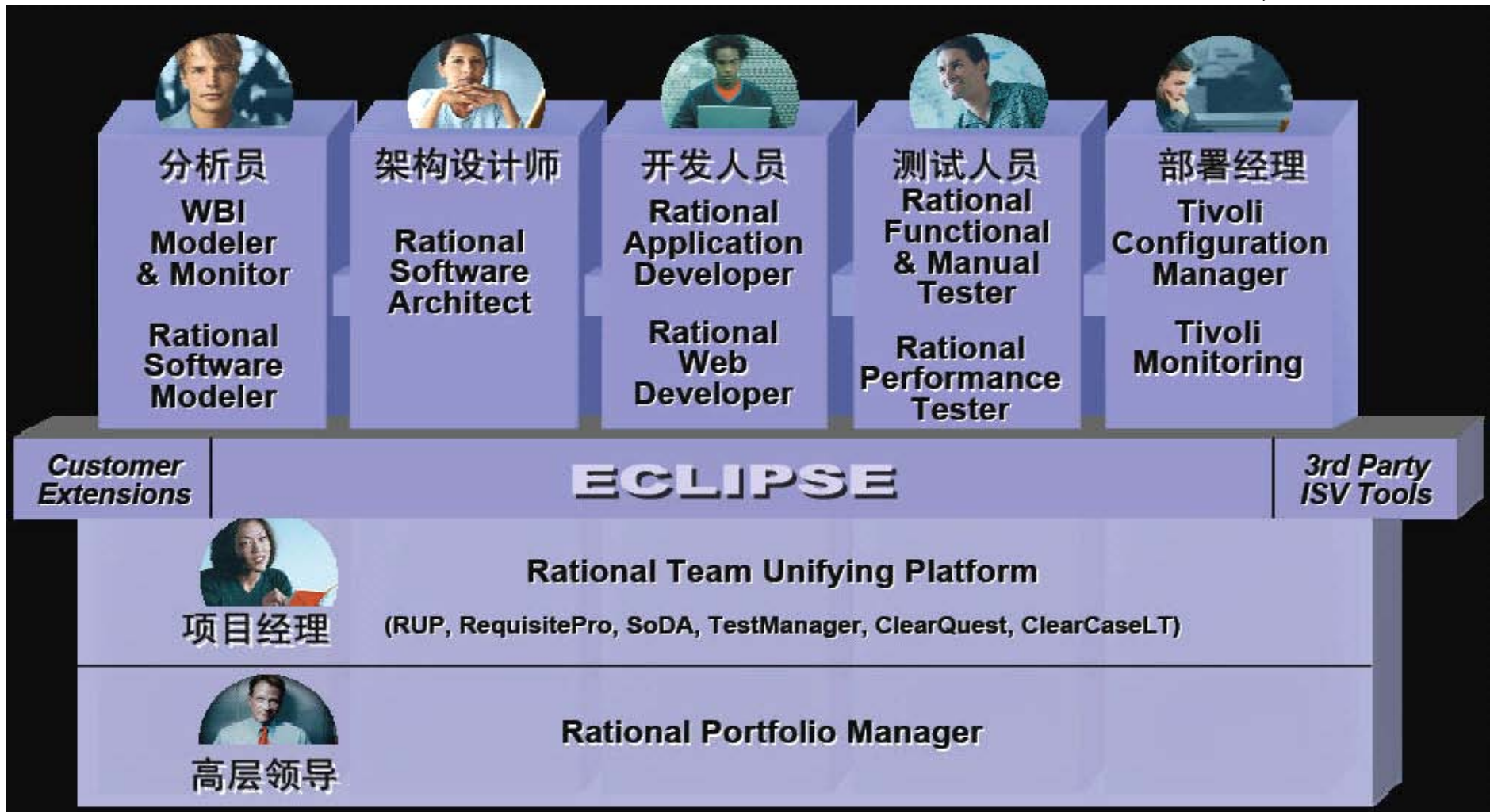


SCM在软件开发中的地位 日显突出

- 📁 软件项目的规模越来越大
- 📁 企业自身对软件开发管理的认识提高
- 📁 实施CMM/CMMI所必须
- 📁 软件外包

软件工程大会

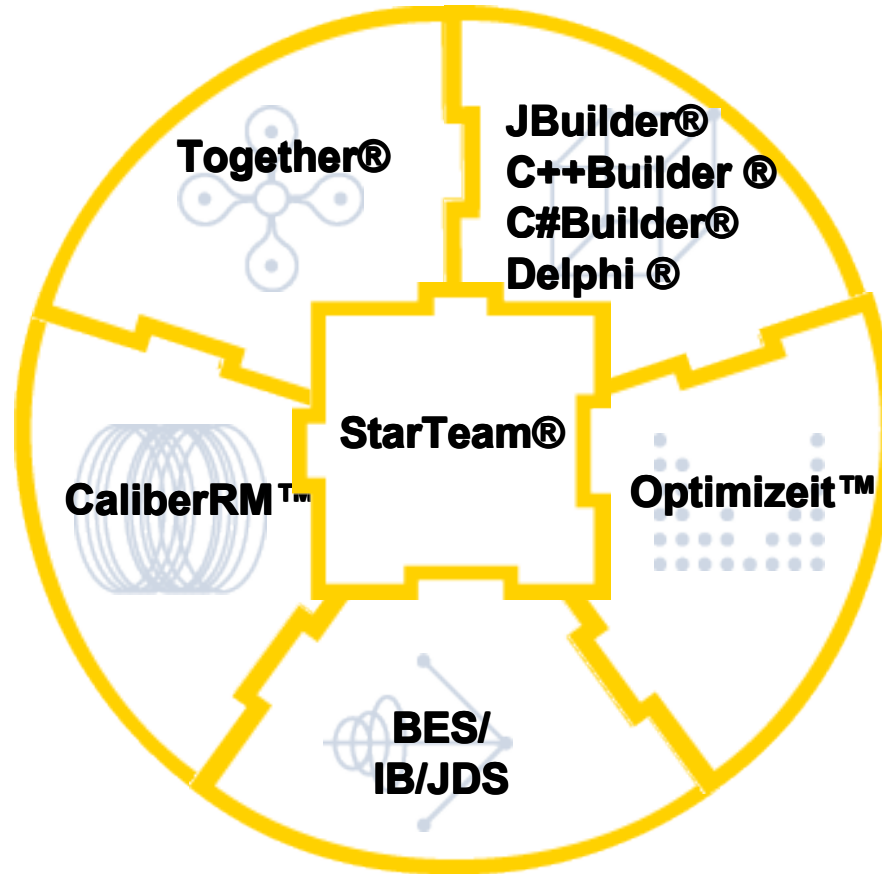
厂商动作: IBM



ALM SDO CoreSDP

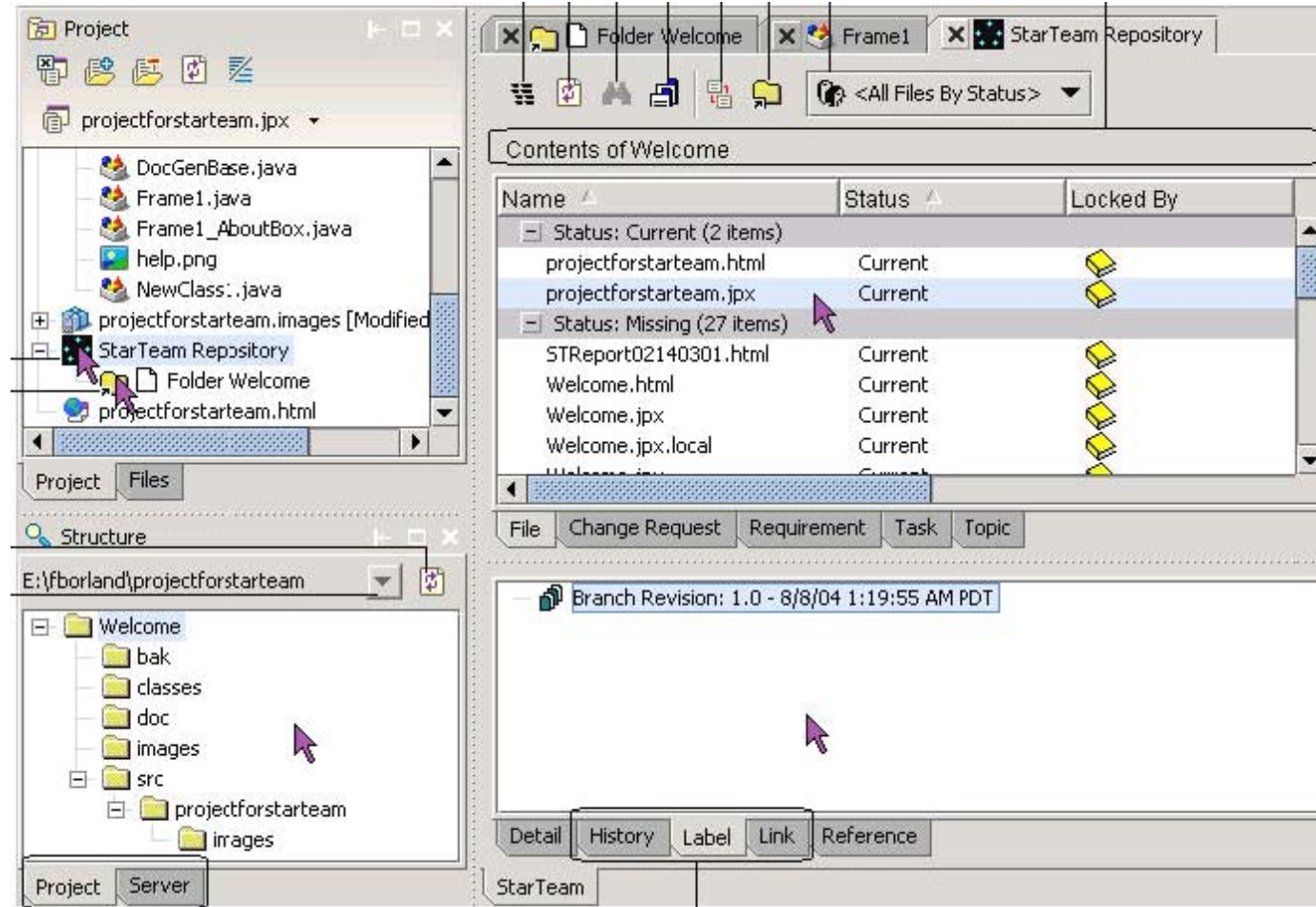
完整性

集成性



软件工程大会

厂商动作: Borland



国内SCM应用现状存在很大改进空间

- 没有专职的SCM人员；没有兼职的SCM人员；没有SCM人员
- SCM人员地位不高；团队整体SCM意识不够
- 对SCM缺乏正确全面的认识
 - 1、范围
 - 2、偏见
 - 1)、过分强调过程与文档，太繁琐
 - 2)、解决不了最终问题

国内SCM应用现状存在很大改进空间

- 实施大多以失败告终
 - 1、缺少中高层支持
 - 2、缺少必要的培训与支持，团队对实施的目标认识不清；
 - 3、不够敏捷
 - 4、没有把握好流程、人员、工具三者之间的关系
 - 5、没有做好配置审计、报告环节

2.配置管理实施规划

- ☞ 启动SCM实施团队，获取资源
- ☞ 评估企业配置管理应用现状
- ☞ 配置管理工具选型
- ☞ 人员培训
- ☞ 制定配置管理计划，定义过程
- ☞ 选定试点项目
- ☞ 全面实施

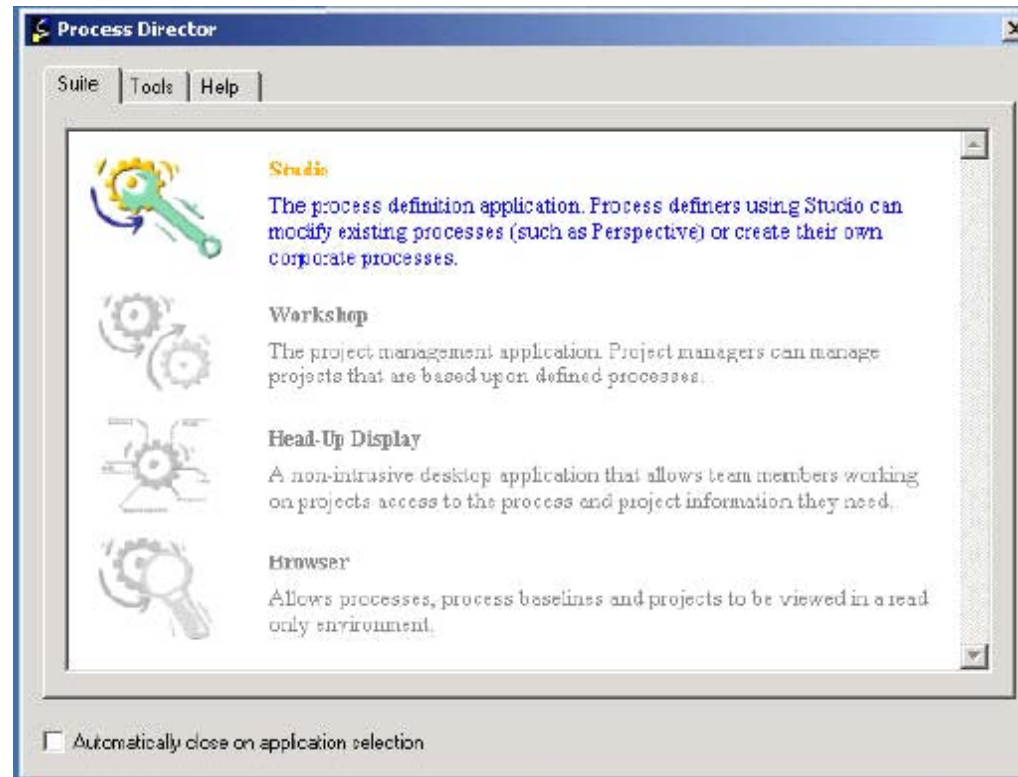


项目软件配置管理现状调查表

项目名称	实施经理	填表人	填表日期
实施状态			
1 项目是否需要实施配置管理	<input type="radio"/> 是	<input type="radio"/> 否	<input type="radio"/> 不清楚
2 项目是否实施了配置管理	<input type="radio"/> 是	<input type="radio"/> 否	<input type="radio"/> 不清楚
3 项目未实施配置管理的原因	<input type="checkbox"/> 没必要	<input type="checkbox"/> 时间不够	<input type="checkbox"/> 人员不足 <input type="checkbox"/> 其他 (有部分需要多人同时)
已实施项目配置管理内容			
4 实施的配置管理的内容包括	<input type="checkbox"/> 版本管理	<input type="checkbox"/> 变更控制	<input type="checkbox"/> 构建管理 <input type="checkbox"/> 配置审计 <input type="checkbox"/> 状态报告 <input type="checkbox"/> 其他
5 是否制定了配置管理制度	<input type="radio"/> 是	<input type="radio"/> 否	
6 是否开展过配置管理培训	<input type="radio"/> 是	<input type="radio"/> 否	
7 是否指派专人负责配置管理	<input type="radio"/> 是	<input type="radio"/> 否	
8 是否制定了配置管理计划	<input type="radio"/> 是	<input type="radio"/> 否	
9 是否要求开发人员必须check out/check in	<input type="radio"/> 是	<input type="radio"/> 否	
10 修改的文档Check in的频率	<input type="radio"/> 每天	<input type="radio"/> 定期	<input type="radio"/> 随意
11 如何保证变更文档按期check in	<input type="checkbox"/> 专人监督	<input type="checkbox"/> 个人自觉	<input type="checkbox"/> 其他 (整体项目计划)

指标项 \ 工具		VSS	StarTeam	FireFly	Rational Clearcase	Telelogic CM Synergy
版本标识	支持对单个文件及目录的版本标识	支持	支持	支持	支持	支持
	自动完成标识目录下所有文件的版本标识	支持	支持	支持	支持	支持
	自动标识文件及目录的版本	支持	支持	支持	支持	支持
	能自定义文件及目录的版本名称	支持	支持	支持	支持	支持
	自定义版本名称的设置无限制	支持	支持	支持	支持	有限制
版本变更	自动判别变更	支持	支持	支持	支持	支持
	增量传输	支持	支持	支持	支持	支持
	变更内容支持配置项修改、添加、删除、重命名	不支持重命名	不支持重命名/删除	支持	支持	支持
	变更记录完整[Who,Why,When,What]	支持	支持	支持	支持	支持
	查询变更记录	支持[可按时段/用户]	支持多种条件查询	支持	支持	支持
版本分支	支持对任一版本文件及目录的分支功能	支持	支持	支持	支持	支持
	分支的层数不受限制	支持	支持	支持	支持	支持
	能记录主版本及各分支版本之间的依赖关系	支持	支持	支持	支持	支持
版本归并	支持对文件及目录不同分支版本的归并功能	不支持目录归并	支持	支持	支持	支持
	版本归并时, 自动检测并显示版本差异	支持	支持	支持	支持	支持
版本跟踪	记录并提供配置项版本变化的历史信息	支持	支持	支持	支持	支持
	提供版本树功能	不支持	支持	支持	支持	支持
	能分析不同版本配置项之间的依赖关系	不支持	支持	支持	支持	支持
	能提供某一配置项版本变化对其他配置项的影响分析报告	不支持	支持	不支持	支持	支持

- 信息的发布渠道: **WiKi,Blog**
- 流程定义与发布工具: **ProcessDirector,RUP Modeler**



- 岗位职责：
 - 1、建立并维护公司配置管理规划、流程和策略；
 - 2、制定并改进缺陷跟踪规划及流程，并与研发、测试和质管人员紧密合作；
 - 3、公司配置管理环境的日常运行维护及系统优化；
 - 4、主动推进公司配置管理工作；
 - 5、参与公司产品及体系认证与维护工作；
 - 6、参与公司质量管理工作；
 - 7、对公司员工进行配置管理概念、流程和工具的培训。

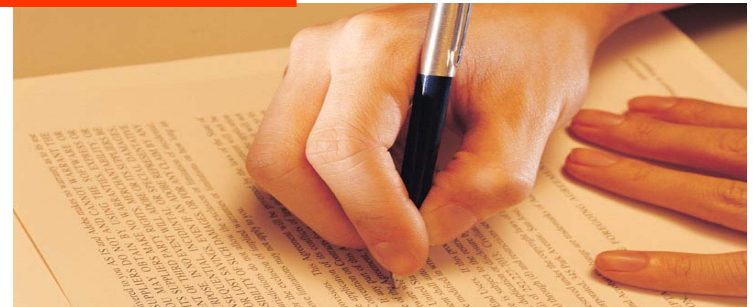
任职要求：

- 1、计算机及相关专业本科以上学历；
- 2、英语4级以上；
- 3、一年以上软件开发经验，两年以上配置管理经验，熟悉VSS、PVCS、ClearCase等工具者优先；
- 4、掌握RUP和CMM原理，有较强的动手能力，能够快速解决配置管理中的各种相关问题；
- 5、熟悉ISO 9000质量管理体系；
- 6、了解CC标准和信息安全基本原理；
- 7、善于沟通、有较强的团队合作精神和协作



3.配置管理概述

1. 软件配置管理的概念
2. 软件配置管理活动



- 表述一：协调软件开发使得混乱减到最小的技术叫做软件配置管理，它是一种标识、组织和控制修改的技术，目的是使错误达到最小并最有效地提高生产效率。
- 表述二：软件配置管理（SCM）是一组活动，被设计来通过标识可能变更的工作产品、建立它们之间的关系、定义管理这些工作产品的不同版本的机制、控制施加的变更以及审计和报告所发生的变更而控制变更。



- 配置标识
- 配置的变更控制
- 配置的状态统计与报告
- 配置的审计与评审
- 构建与发布管理

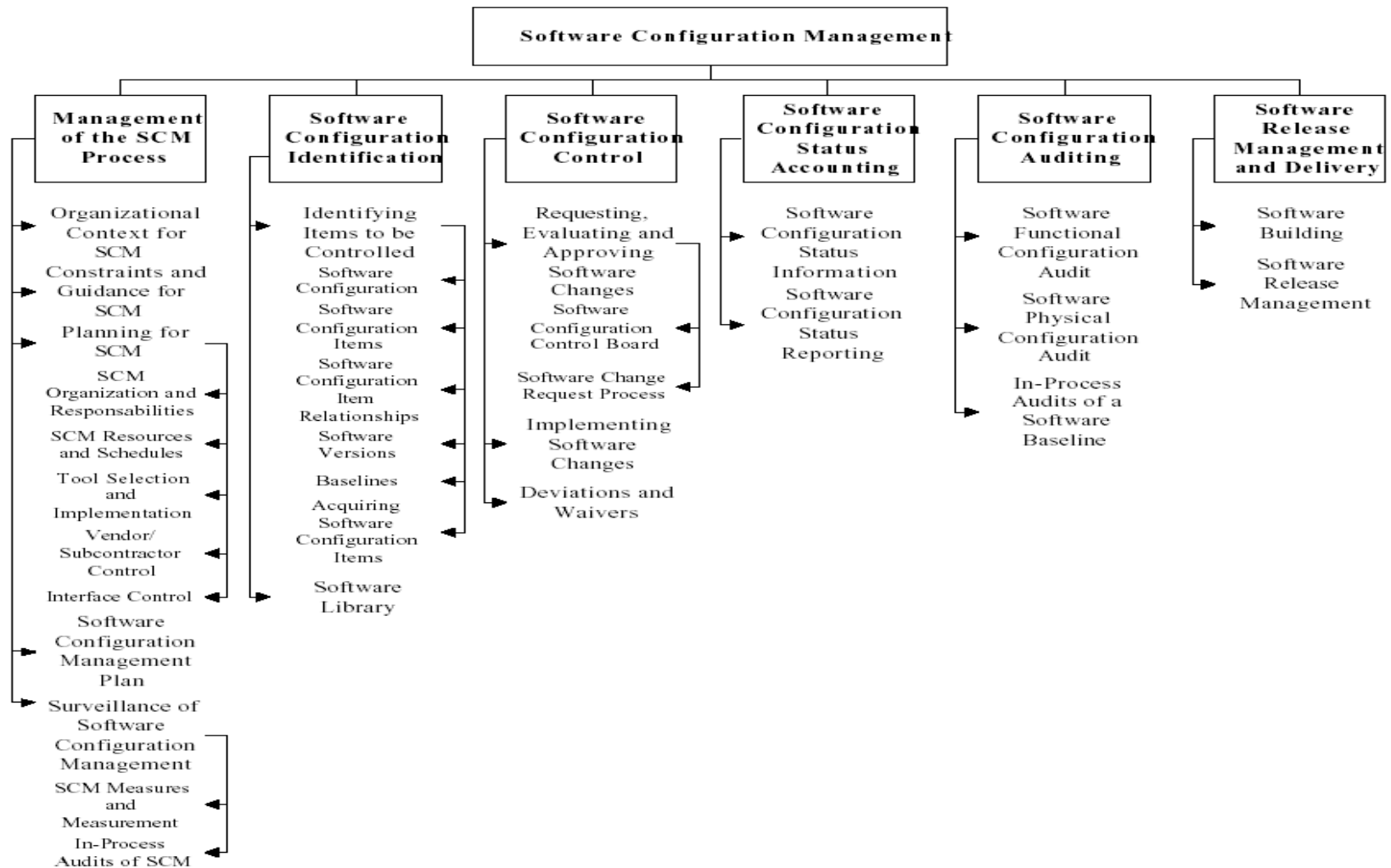
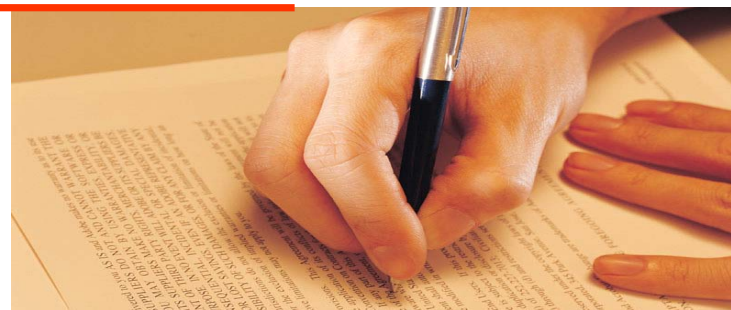


Figure 2 Breakdown of SCM Topics

4.融入敏捷思想

- 📁 敏捷配置管理特征
- 📁 搭建敏捷配置管理环境
- 📁 持续集成、构建管理、发布管理
- 📁 敏捷变更管理
- 📁 建立模式文化



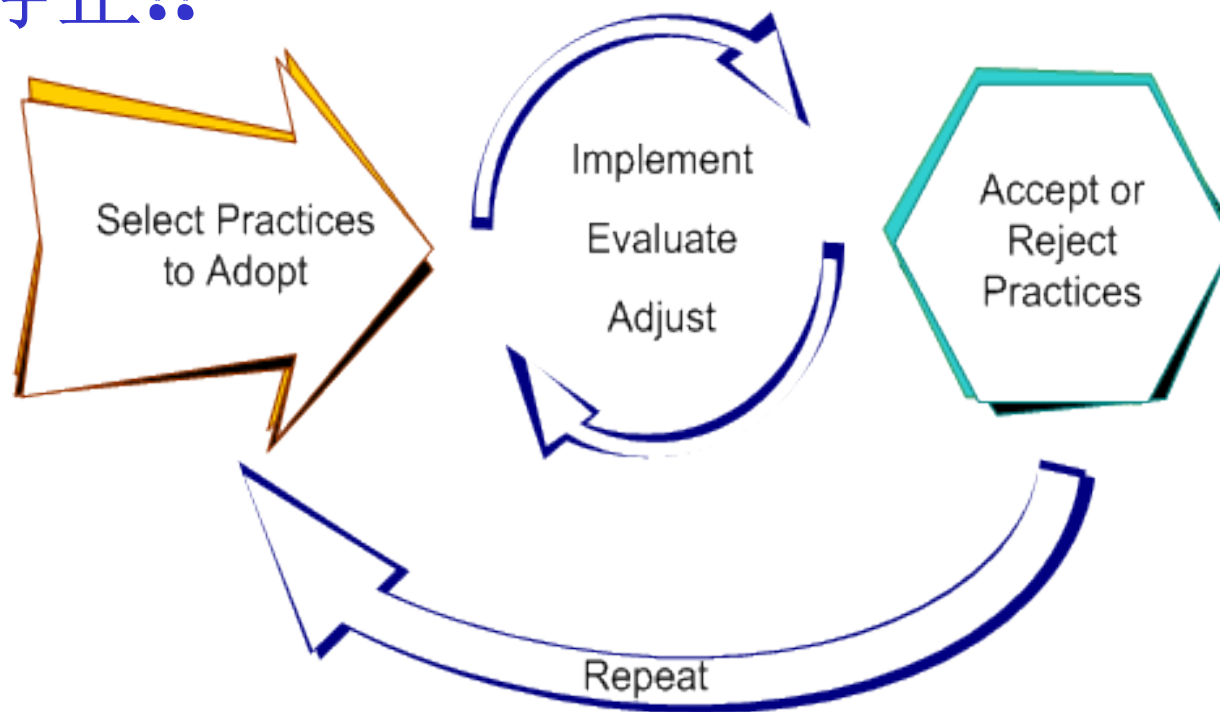
软件工程大会



敏捷开发实践

- 敏捷团队所遵循的过程
- 程序员和管理者已经使用这些实践多年甚至上十年
- 已经被敏捷专家广泛采纳、研究并撰写成文章、书籍

- 一次采纳一到两个实践
- 不要给团队带来不需要的关注点
- 不要停止!!



■ 当前最糟糕的问题

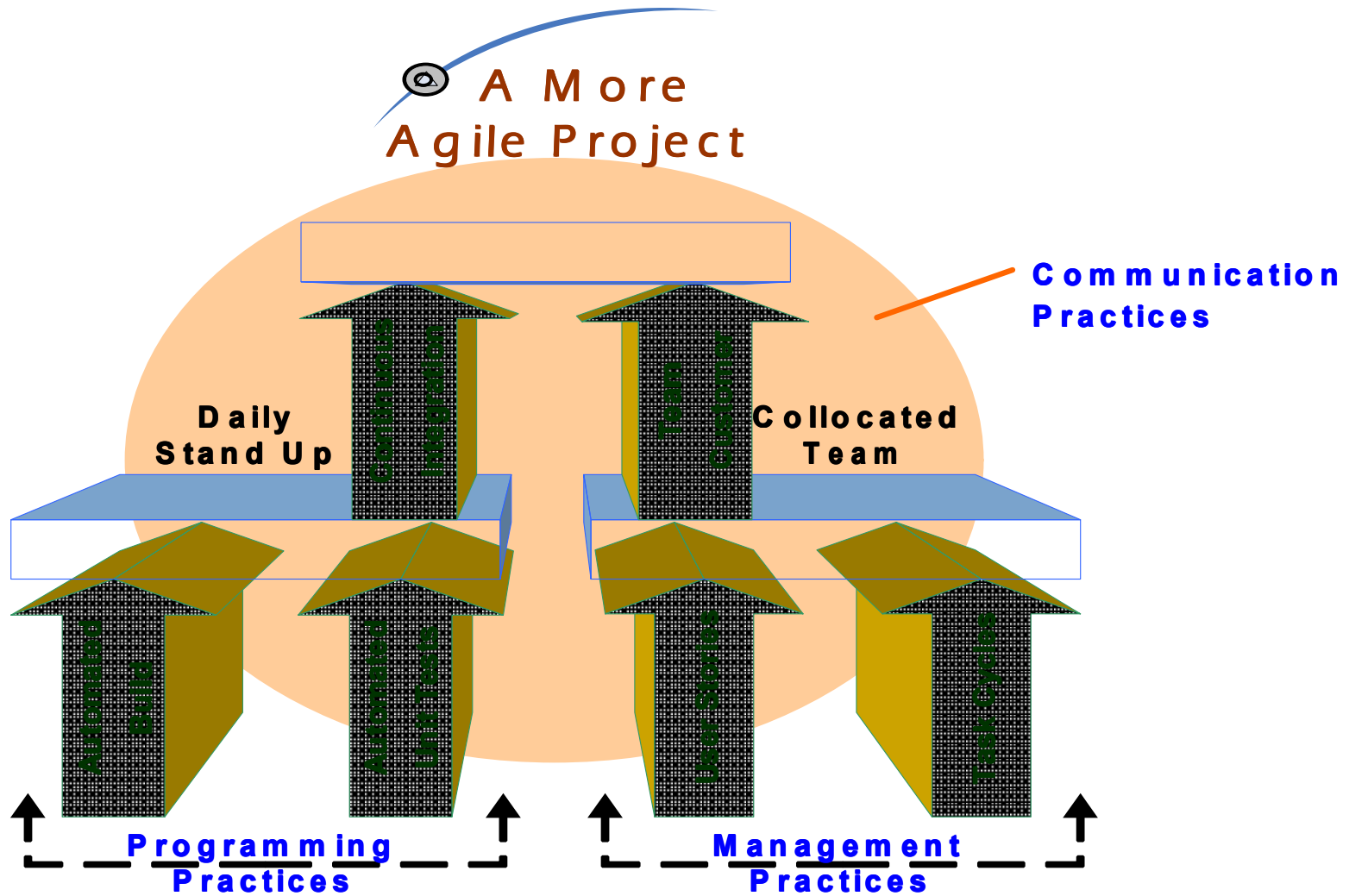
- 找出你的项目中所面临的最糟糕的问题
- 将它与敏捷实践中对适合解决此问题的实践进行配对
- 从相关的机构中购买支持
- 当这一问题不再是你的最糟糕问题时，重复上述循环

■ 对反对者具有最大震撼效果

- 找出对于你的项目来说，实现的快速性和**RIO**（投资回报率）综合起来最好的敏捷实践
- 解决的应该是当前所面临的最糟糕问题
- 从快速、敏捷、成功的实践采纳中受益
 - ◆ 鼓舞士气，赋予团队力量
 - ◆ 说服那些持消极或反对态度的人

- 先作最容易的事情，困难的放在后面
 - 首先尽早地打一次胜仗
 - 接下来，使用这一成功解决团队的最糟糕问题
 - 请记住，最糟糕的问题也是最难解决的问题

- 并不是每一个实践都适合每一个项目
- 坚持采用一个失败的实践是使得项目成本高昂的原因之一
- 当放弃某个实践时，不妨问问：
 - 为什么它不适合我们的项目？
 - 我们还有必要再尝试这个实践吗？



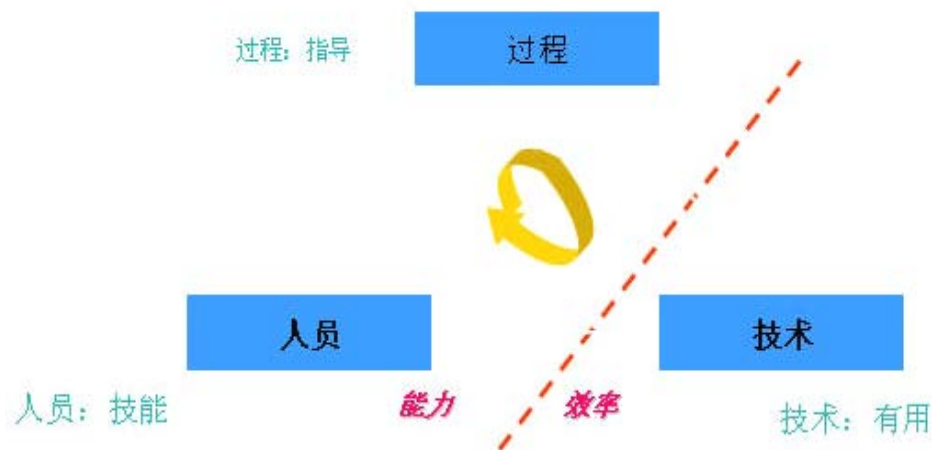
软件工程大会

- 这一采纳敏捷实践的过程同样也适用于敏捷配置管理!!!

- 无摩擦力的代码集成
 - 自由修改私有工作空间中的代码
 - 离线工作
 - 轻松、快速地与仓库同步代码
 - 轻松、快速的分支合并
- 支持重构
 - 文件、目录的重命名、删除、修改位置等
- 持续集成
 - 避免出现“它在我的机器上能工作”场景

- 自动化的构建
 - 开发人员构建
 - 发布构建
 - 集成构建
- 简单与弹性
- 低成本管理

- **SCM**是整个团队的责任
- 响应变化而不是控制它
- 简单、紧凑的配置标识与报告
- 协作与自动化



■ 构建管理

- 开发人员构建
- 发布构建
- 集成构建

■ 自动化的构建

- 可能是**ROI**最快的
- 实现敏捷的构建管理使得
 - 只需要点击一下按钮就完成整个应用的构建
 - 整个应用的构建少于10分钟
- 好处包括
 - 减少重复性工作
 - 避免出现瓶颈
 - 捕获编译和集中性的问题
 - 是其他敏捷实践的基础

- 具有可靠的**ROI**，但需要作出更多的努力
- 为系统实现一个完整的测试集合，在每次构建后自动运行
- 好处
 - 尽早地发现缺陷
 - 不断增加对代码的自信



- **XP**所强调的实践之一
- 以自动化的测试、自动化的构建为基础
- 先实现每日构建，然后实现持续集成

- 在一个五十人的团队、二十万行代码的项目中，我们每天要集成二十多次。
- 选择运行主创建的机器。我们选择了一台叫做“投石车”的计算机（我们经常玩“帝国时代”J），这是一台装有四个CPU的服务器，非常适合专门用来做创建。（由于完整的创建需要相当长的时间，所以这种马力是必须的。）
- 关键是要让所有的事情都完全自动化，并且要经常进行集成，这样才能尽快发现错误。然后，人们可以随时修改需要修改的东西，因为他们知道：如果他们做的修改引起了集成错误，那也是很容易发现和修补的。



软件工程大会

来自ThoughtWorks公司的故事

Atlas Build Logs

Today

- [Aug 16, 06:14 PM](#)
- [Aug 16, 05:30 PM](#)
- [Aug 16, 04:18 PM](#)
- [Aug 16, 02:52 PM](#)
- [Aug 16, 01:14 PM](#)
- [Aug 16, 12:26 AM](#)

Current Build Started At

07:03 PM

BUILD COMPLETE

Build 452.7 - Aug 16, 05:30 PM

Functional Tests

Summary

[1 File Revisions](#) [0 bugs addressed](#) [0 Test Failures](#) [Build Log](#)

File Revisions

Modified By	Modified On	File	Comment
Junshu Zhou	Aug 16, 05:11 PM	ChargeGenTest.java	

Bugs Addressed

No bugs addressed

Test Failures

■ 发布管理



■ 发布管理



- 变更管理是用来控制和跟踪项目或产品中所发生的变更行为，以保证项目或产品与客户的期望相一致。
- 敏捷变更管理则是在此基础上不断增加项目的响应能力和快速接受变更的能力
- 首先端正心态
- 采纳敏捷开发实践：自动化的构建、客户现场、重构、结对编程.....
- 精简工件，以降低变更跟踪的复杂性

- 模式：每一个模式描述了在我们周围不断重复发生的问题，以及该问题的解决方案的核心。这样，你就能一次又一次地使用该方案而不必作重复劳动。
- 模式四要素
 - 1、模式名称
 - 2、问题
 - 3、解决方案
 - 4、效果

- 模式的好处
 - 1、避免重复劳动
 - 2、团队交流的共同语言
 - 3、站在前人的肩膀上
 - 4、不断积累的财富



- 设计模式、分析模式、企业应用架构模式、**JAVA**模式、**J2EE**模式、**DELPHI**模式.....



软件工程大会

■ 软件配置管理模式!!!



- 主线、活动开发线、私用工作区、任务级提交、码线策略、单元测试、回归测试、任务分支.....

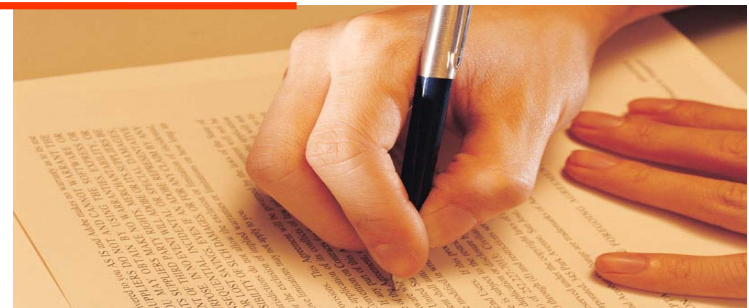
4.主要参考文献

📖 《软件配置管理模式》 Stephen
P.Berczuk,Brad Appleton 著 中国电力
出版社

📖 Characteristic of the Agile SCM
Solution

📖 The Future of Agile CM

📖 Making SCM Agile



软件工程大会

谢谢！

