



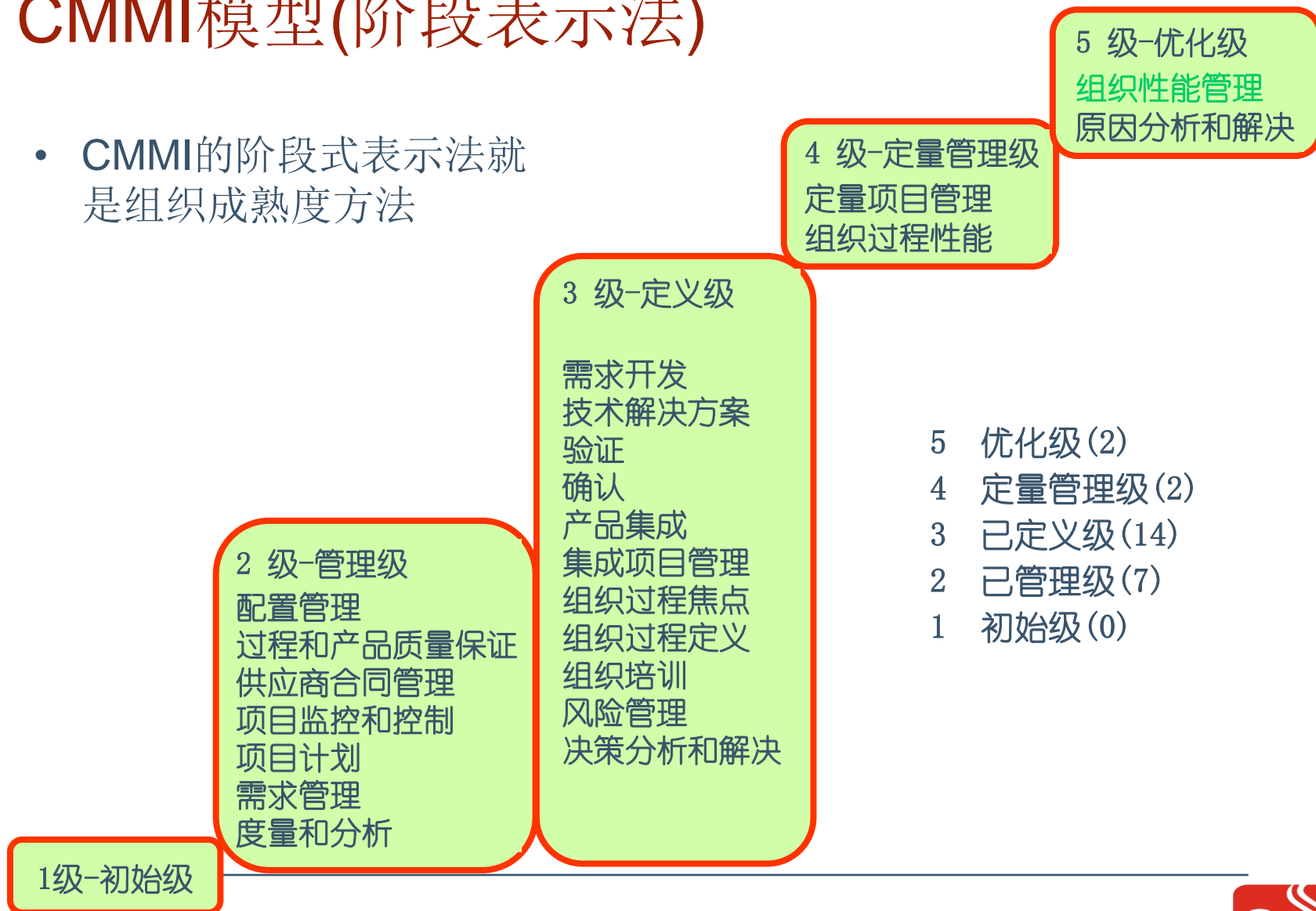
CMMI各过程域的体系建立要点

Jacky yi
Lead consultant
Share-Win Technology



CMMI模型(阶段表示法)

- CMMI的阶段式表示法就是组织成熟度方法





CMMI模型(持续表示法)

级别\分类	Engineering (6)	Project Management (6)	Process Management(5)	Support(5)
2级 受管理级 Managed (7)		PP(项目计划) PMC(项目监控) SAM(分包合同管理) REQM (需求管理)		CM(配置管理) PPQA (过程和产品质量保证) MA(度量与分析)
3级 已定义级 Defined (11)	RD(需求开发) TS(技术解决) PI (产品集成) VER(验证) VAL(确认)	IPM(集成项目管理) RSKM(风险管理)	OPD(过程定义) OPF(过程聚焦) OT(培训)	DAR (决策分析与解决方案)
4级 定量管理级 Quantitatively Managed (2)		QPM(定量项目管理)	OPP(组织过程性能)	
5级 持续优化级 Optimizing (2)			OPM(组织性能管理)	CAR (因果分析和解决方案)



项目分类

- 项目分类
 - 产品：自主开发，没有合同
 - 项目：和客户有合同
 - 研究：研究新技术，新产品，风险很大，可预测性，一般人员规模小
 - 维护：已有系统的维护
- 区别对待
 - 立项流程不同
 - LCM不同
 - 管理的严格程度不同
- 反应在公司的方针里面



项目立项

- 立项的条件
- 立项的流程
- 撤项的条件
- 撤项的流程
- 是否2阶段立项?
 - 可行性研究和需求开发的立项
 - 开发立项
- 类型不同是否立项流程不同
- 规模不同，是否立项的流程不同
 - 1000个FP
 - 2000个FP



需求工程



操作概念描述的要点

- 操作概念的定义:
 - 系统的全生命周期的业务/系统用例的集合
- 描述要点:
 - 识别了用户角色
 - 描述了用户的应用场景和用例
 - 描述了用户角色和用例之间的关系
 - 描述了系统的运行环境
 - 从全生命周期的角度识别了系统的功能
- 可操作概念主要体现在2个文档中：客户需求，产品和产品构件需求（相当于需求规格说明书）。



确认需求的方法

- 外部正式确认：客户确认的方式
 - 客户对需求的正式签字确认
 - Q&A、电子邮件等作为客户确认的间接证据
- 内部正式确认
- 编写系统测试用例
- 内部承诺
- 需求得到客户确认后，即形成需求基线



过程定义过程中的注意问题

- 需求工程要分为2个过程来写
- 需求开发
 - 需求开发计划制定
 - 需求获取
 - 识别合适的需求提供者
 - 和需求提供者对需求达成一致
 - 需求分析与描述
 - 对于外包类的企业,如果是从设计开始或者是从编码开始,则获取与分析描述的活动是和甲方的沟通需求
 - 需求验证与确认
 - 需求评审
 - 系统测试用例的编写
- 需求跟踪矩阵的建立



需求工程要分为2个过程来写

- 需求管理
 - 需求跟踪矩阵的变更
 - 可以不单独写流程,融合在CI变更流程中
 - 矩阵的建立
 - 需求变化了,修改矩阵
 - 设计完成后,要修改矩阵
 - 测试用例完成后,要修改矩阵
 - 代码完成后,
 - 需求变更的流程
 - 并非所有的需求变更都走同样的流程
 - 需求变更流程建议不要和CI的变更流程合成一个
 - 需求变更的申请单与CI变更的申请区分开



需求开发的最小文档集-1

- 需求开发过程定义
- 需求管理过程定义
- 有些实践可以合并到需求开发中去
- 客户需求说明书 REQM sp1.1,REQM sp1.2
- 软件需求规格说明书 RD sp2.1,RD sp2.2,RD sp2.3,RD SP3.1,TS SP1.2,RD SP3.2,RD SP3.3,RD SP3.4
- 需求跟踪矩阵 REQM sp1.4
- 需求评审检查单(专家使用)
- 初始需求检查单 RD sp1.5
- 需求说明书检查单



需求开发的最小文档集-2

- 其他说明
 - 需求开发计划可以合并到项目开发计划中 REQM gp2.2
 - 需求评审记录在V&V中定义,但是注意要包括需求承诺
REQM sp1.2,RD sp3.5
 - 需求不一致检查表可以在PPQA中定义 REQM sp1.5



需求文档

- 需求文档一般分2个：
 - 用户（客户）需求说明书，在模型中对应为客户需求
 - **SRS**：在模型中对应为产品和产品构件需求
- 这2个文档对于小项目(不超过9个人月)可以合成为1个
- 这2个文档应该都在需求阶段做。在瀑布模型中的阶段是需求分析阶段，实际上该阶段更准确的叫法是需求开发阶段，其任务包含了**RD PA**中的活动。
- 要注意的是这种划分并不绝对，客户需求也可能是在需求分析的前一阶段形成，如在项目立项阶段或者叫可行性研究阶段，也可能跨了2个阶段



客户需求说明书中的主要内容

- 企业简介
- 组织结构
- 角色划分
- 旧系统的业务流程
- 运行环境
- 旧系统存在的问题
- 新系统的目标
- 新系统的范围
 - 部门覆盖
 - 地理覆盖
 - 业务覆盖
- 新系统的作业流程
- 功能需求:要划分优先级
- 非功能需求
- 接口需求
- 约束条件

•对于非信息系统类的软件在上述基础上裁剪.



产品需求规格说明书

- 系统目标
- 需求分解(RD SP2.2,RD SP3.2)
- 功能需求
- 业务逻辑
- 需要处理的数据
- 界面需求
- 接口需求(RD SP2.3)
- 需求优先级(RD SP3.3)
- 约束条件
- 其他非功能性需求的描述
- 系统验收的准则(尽可能量化,明确)



需求跟踪矩阵建立的基本原则

- 必须的矩阵
 - 客户需求与产品需求的跟踪
 - 产品需求与测试用例的跟踪
- 100%的接口需求需要建立客户需求-产品需求-设计-编码-测试用例的跟踪矩阵
- 性能需求可以不建立跟踪矩阵
- 影响系统架构的功能需求要建立全部的跟踪矩阵
 - 关键需求要建立跟踪矩阵
 - 核心需求要建立跟踪矩阵
 - 全局性需求要建立跟踪矩阵
- 横向跟踪矩阵根据实际情况自己来定是否需要



技术解决方案



技术解决方案需要的最小文档集-1

- (1) 项目方案选择准则模板: TS sp1.1
- (2) 产品构件解决方案: TS sp1.1
- (3) 设计规范
 - 概要设计说明书编写规范: TS sp2.1, TS sp2.3
 - 详细设计说明书编写规范: TS sp2.1
 - 数据库设计规范
 - 命名规范
 - 界面设计规范: TS sp2.1
 - 设计评价准则
 - 接口评价准则
- (4) 制作购买复用分析结果: TS sp2.4
- (5) 各种编码规范: TS sp3.1 (源代码)
- (6) 用户手册编写规范: TS sp3.2
- (7) TS 过程定义



概要设计文档的内容

- 功能模块划分
- 与其他系统的接口（协议、标准）
- 模块之间的内部接口
- 数据库模型（ER图）
- 技术路线（架构）
- 界面设计(至少要定义UI的风格等基本原则)
- 类（职责定义和类间关系）



详细设计文档的内容

- OO方法:
 - 交互图
 - 类图（细化到属性）
 - 属性
 - 操作的具体实现方法
 - 对外的接口
- 结构化方法
 - 功能设计
 - 数据结构
 - 算法设计
 - 输入
 - 输出
- 界面设计
- 接口设计



设计评价准则

- 概要设计:主要考虑影响系统架构的需求
- 无二义性（术语不清楚）
- 高效开发（可复用的构件、技术难度低、人员要求低）
- 高内低耦合
- 实现的成本
- 易于分工
- 可实现性
- 可维护性
- 可扩展性
- 对于OO的设计评价准则
 - SRP,LSP,DIP,OCP,ISP



接口设计描述规范

- 接口名称
- 调用方式
- 传输的数据:
 - 哪些数据项?
 - 数据格式?
 - 长度?
 - 类型?
- 频度（实时/批处理）
- 数据同步机制
- 平台无关性
- 遵循的标准
- 提供的功能



接口设计的评价准则

- 数据量的多少
- 频度
- 耦合度
- 适应性
- 功能单一性



技术解决方案需要的最小文档集-2

- 其他说明
 - 技术数据包: **TS sp2.2**。对于软件开发、简单的开发不需要单独的文档



数据库设计指南

- 数据库命名规范
 - 实体（表）的命名
 - 属性（列）的命名
 - 视图的命名
 - 触发器的命名
 - 存储过程名
 - 变量名
 - 命名中其他注意事项
- 字段设计原则
- 键选择原则
- 索引使用原则
- 数据完整性设计
- 其他设计技巧



界面设计指南

- 布局
- 颜色
- 文字
- 大小
- 热键
- 各控件的风格



编码风格规范

- 书写的风格
- 注释的风格
- 控制结构的风格
- 处理的风格
- 事件驱动的编程风格
- 其他忠告

• 参见<程序设计风格>专题培训PPT



用户手册编写指南

- 有快速入门
- 有Q&A、FAQ
- 有运行环境的描述
- 保持术语一致性
- 要有索引
- 要有联机Help
- 要有文档内容的链接
- 区分读者群,不同的读者内容不同
- 组织排版格式,字体等要求漂亮一些
- 文档可以复用
- 多用图表



TS过程描述

- 按正常的设计与编码过程描述
 - 概要设计
 - 详细设计
 - 编码
 - 单元测试
 - 编写用户文档
- 注意不要遗漏活动
 - 技术方案的选择
 - 制造购买复用的分析



产品集成



接口与用户界面的概念区分清楚

- 广义上用户界面是接口的一种,狭义上是不同,接口是软件和软件,软件和硬件,硬件和硬件之间的,不是人和软件之间的.
.PI中的接口概念是狭义的概念.
- 接口的需求描述重点在于定义双方需要交换什么信息,而不注重具体的手段,如,如果要给汽车发一个加速的信号,不管你是通过手柄控制,还是脚踏,或者摁按钮。双方的接口是需要人给汽车一个加速信号。



产品集成的最小文档集-1

- 1 PI过程定义
- 2 产品集成规程（项目级）
 - 集成顺序 sp1.1
 - 集成环境的要求 sp1.2
 - 集成步骤 sp1.3
 - 集成的进入与退出准则 sp1.3
 - 单元测试结束的准则
 - 集成测试结束的准则
- 3 交付清单 sp 3.4



产品集成的最小文档集-2

- 其他说明
- 接口描述 sp2.1 sp2.2 反映在客户需求，产品需求与产品构件需求文档中就可以，不需要单独的文档，参见RD PA
- 接口评审记录 sp2.1 反应在概要设计或详细设计文档中就可以，不需要单独的文档，参见TS PA
- 接口设计文档 sp2.2 反应在概要设计与详细设计文档中就可以，不需要单独的文档，参见TS PA
- 接口测试报告 sp3.1 反应在产品集成测试报告中或者单元测试报告中就可以，不需要单独的文档，参见VER PA
- 接口测试记录 sp3.3 反应在产品集成测试记录中就可以，不需要单独的文档，参见VER PA
- 产品集成计划 gp2.2 gp2.3 gp2.4 gp2.5 gp2.7可以反应在项目计划中，参见PP PA。
 - 需要的资源（人，设备，工具等）
 - 人员的职责
 - 培训需求
 - 集成活动的任务
 - 人员参与集成活动的时间



验证与确认



验证与确认的概念

	验证Verification	确认Validation
定义	Verification confirms that work products properly reflect the requirements specified for them.	Validation confirms that the product, as provided, will fulfill its intended use.
重点	做法是否正确，强调过程的正确性 verification ensures that “you built it right;”	结果是否正确，强调结果的正确性 validation ensures that “you built the right thing.”
CMMI中的目的	确保所选择的工作产品满足指定的需求	当产品或者产品组件被置于其要求环境中时，产品或者产品组件能够完成其所期望的功能。
参照物	上一阶段的输入	原始需求
可采用的方法	代码走查、审查、测试和正确性证明	审查、测试和正确性证明



V&V的手段

- 同行评审
 - 走查
 - 审查
 - 技术复审
- 测试
 - 单元测试
 - 集成测试
 - 系统测试
 - 验收测试
- 模拟
- 原型



V&V的最小文档集

- V&V的策略
 - 对哪些工作产品,在什么时间,做什么类型的V&V活动
- 测试的流程定义
 - 各种测试结束的准则
 - 单元测试,集成测试,系统测试,回归测试
- 评审的流程定义
 - 评审准备的充分性如何度量
 - 评审充分性如何度量
- 缺陷跟踪的流程定义
 - 可以和PPQA的缺陷跟踪流程合并
- 缺陷分析的流程定义
- 其他V&V流程定义(如果有的话)
- 缺陷记录的模板
- 测试用例的模板
- 测试计划的模板



同行评审准则 - 1

- 对需求文档进行同行评审时，要规定：
 - 评审覆盖率应为100%
 - 要有明确的评审检查单
 - 对评审发现的问题进行跟踪
- 对设计文档进行同行评审时，要规定：
 - 评审覆盖率应为100%
 - 要有明确的评审检查单，在检查单中要明确要求考虑程序结构的形态（调用宽度，调用深度，内聚度，耦合度，结构化程度）
 - 对评审发现的问题进行跟踪
- 对代码进行同行评审时，要规定
 - 新编代码的关键部位和关键算法要100%进行同行评审
 - 非新编代码采取采样评审，采样比不少于25%
- 进行同行评审的模块及其占总规模的百分比（不应小于30%）
 - 要有明确的评审检查单



同行评审准则 - 2

- 对评审的过程给出准则，并进行必要的度量，例如要规定：
 - 同行评审工作量占该阶段总工作量应逐步提升到**15%**以上，到高成熟度（例如**CMMI L3**）时，逐步过渡到**25%**以上
 - 总的评审工作量应逐步提升到大于总的测试工作量
 - 评审准备时间和评审会议时间之比一般应大于**1**
 - 评审准备期间发现的缺陷数和评审会议期间发现的缺陷数之比一般应为**2:1**
 - 评审效率与测试效率之比应逐步提升到**3:1**
 - 做好同行评审前的准备工作，评审人员合格、有时间准备、准备工作分工明确、具体



配置管理 CM



数据管理与配置管理

- 数据管理的内容
 - 识别数据: 要管理哪些数据?
 - 内容定义: 应该包含哪些章节, 每个章节应该如何写。
 - 形式定义: 是WORD, 还是PPT, 是表还是文字等
 - 权限管理: 谁有权力读取?
- 配置管理的内容
 - 识别CI
 - 定义基线
 - 管理CI的变更
 - CI的状态报告
 - 基线的审计
- 对这些数据能识别出来具体的文档的, 就识别出来具体文档, 不能识别出具体文档的, 就识别出类别来, 然后按找项目数据管理表来定义他们的属性, 对他们进行管理。比如在做计划的时候无法识别出来将来会发生那些邮件的往来, 但是可以识别出这些类来, 按类来管理, 这种管理是粗放式管理



纳入基线管理的CI

- 原则1:所有交付给客户的文档, 代码, 可执行程序, 购买来的可复用构件等必须纳入基线。
- 原则2:影响了对外的承诺的文档
 - 项目的阶段计划必须纳入基线来管理。(对外承诺变了)
- 原则3:其变化影响了其他文档
 - 所有对交付产品有重要影响的文档资料等必须纳入基线。
 - 主动变化, 被动变化
- 原则4:变化的才纳入CM, 其他DM
- 优先级依次降低



识别为CI的注意事项

- 配置项可能不是一个文件,可能是一个文件的多个部分,也可能是多个文件
- 要定义清楚何时纳入配置管理
- 配置标识的唯一性
 - 项目
 - 文档类型
 - 顺序号
 - 版本



CCB的建立

- 变更控制委员会（CCB）：负责评价和批准（或不批准）对配置项所提出的更改，并负责保证那些已批准的更改能得到实施的组。
- 变更控制委员会(CCB)由部门（副）经理或项目责任人或业务经理（统括）、项目负责人，各层项目组长（**Sub Leader**）、项目组内的技术负责人（非项目负责人）、关键的软件开发人员、关键的软件测试人员、PPQA、配置管理员以及其他有关人员组成。
- 由除项目负责人、PPQA、配置管理员以外的人员担任主席。
- CCB至少要有3人：CCB主席、项目负责人、设计人员。
- 根据项目规模和所涉及学科的多少，CCB可以下设一至多个变更管理组，分别处理在各个分组（**Sub Team**）范围内的变更。高级别变更由CCB进行控制。
- 变更控制组（CCT）：变更控制组由各层项目组长（**Sub Leader**），项目组内的技术负责人、关键的软件开发人员、关键的软件测试人员、PPQA、配置管理员以及其他项目相关人员组成。低级别变更由各变更管理组进行管理。



配置管理的核心是分级控制

- 变更级别：分为高、低二级。
 - 高：
 - 影响其他项目分组（**Sub Team**）的变更
 - 影响其他项目或者影响项目外部承诺的变更
 - 单次变更估算规模大于项目总体估算规模**5%**
 - 单次变更导致工作量成本增加超过**1**人周
 - 项目总体累计变更规模大于项目总体估算规模**30%**后的变更
 - 低：其他情况。



物理审计

- 物理审核的目标是验证在配置管理系统中建立基线的工件是否为“正确”版本,可以由CM来完成。
物理审计在产品交付前进行,是按配置管理计划检查:
 - 在当前时间下应入库的配置管理项是否均入库
 - 已入项之间版本关系是否一致等
 - 作出应交付的软件产品是否完整、齐套的结论,
- 以下各项说明为支持物理审核而需要做的工作。
 - 创建应该出现在配置管理 (SCM) 中的项目列表。
 - 检查在 SCM 中维护的项目。
 - 创建一个“差异列表”,表示已在 SCM 中维护的项目以及应该在 SCM 中维护的项目之间的差异。
-



功能审计

- 功能审计的目标是核实软件配置项的实际性能是否符合它的需求。
- 功能审计需要全面掌握测试、故障及其排除等信息后综合推导才能得出结论,要有PM+PPQA+CM来做.
- 功能审计所需要做的工作:
 - 准备一个验证表, 列出所有功能方面的需求, 而且对每个需求都引用测试过程、测试行为的实例、相应的测试结果和/或完整记录需求验证情况的分析和/或演示报告。
 - 核实是否已正确实施了所有变更请求。
 - 核实是否已对软件正确应用了所有更改。
 - 文档差异、建立纠正操作和完成日期。



库的划分-1

- (1) 开发库:
 - 存放开发过程中需要保留的各种信息，供开发人员个人专用，该库中存放的是开发人员自己保存的内容。该库一般不定义正式的管理流程。
- (2) 受控库:
 - 当开发人员在自己的工作开发到一定程度后，认为可以提交测试或者提交给项目经理检查了，他可以提交到受控库，由测试人员或项目经理检查。检查后的文档仍存在受控库中。
- (3) 基线库:
 - 存放纳入到基线中的所有的CI。
 - 基线的变更权限最严格，一般在入基线前要经过QA，验证，审批等活动，写的权限只掌握在一个人手里。
 - 当预期的基线所包括的所有文档在受控库中都达到可以基线化的状态时，可以将这些CI转入基线库中，也可以在受控库中，通过CM工具标记为基线。
 - 基线的变更必须经过CCB的批准或者CCB授权的人员来审批；
- (4) 产品库:
 - 在开发的软件产品完成系统测试之后，作为最终产品存入库内，等待交付用户或现场安装。
 - 这次在产品库中存放的是产品基线。也可以没有该库，在基线库中通过标记来注明，为了保证发布的简便性，一般单独出来产品库。



库的划分-2

- (5) 非纳入配置管理的数据库：
 - 在开发过程中，没有纳入配置管理的那些数据，只是控制权限，对这些数据进行分类存放。
 - 包括了物理的硬拷贝的资料。
- (6) 归档库：
 - 在开发过程中，所有的不再发生变化的文档做为历史可以存放在这个库中。比如当项目结束了，所有的资料都可以归档到这个库中，上边提到的2, 3, 4, 5类库在项目结束都可以纳入到这个库中来，这个库也是组织过程资产库的组成部分。
 - 在归档库中不包括度量数据库。

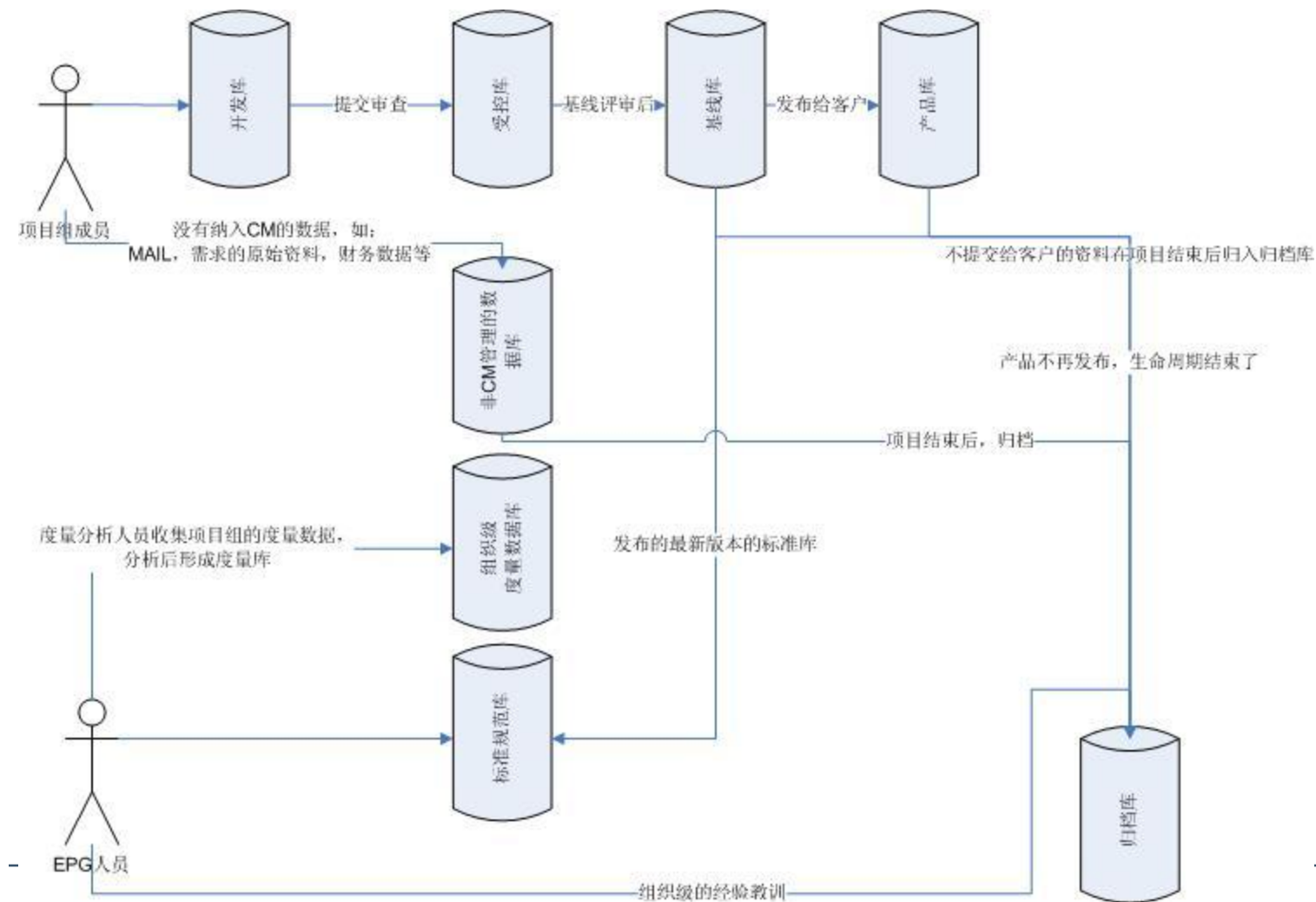


库的划分-3

- (7) 规范与标准库：
 - 存放组织的规范与过程标准，这里应该是保存最新版本的组织的规范与标准，相当于是组织级规范与过程标准的产品库。对于历史版本的组织规范与标准可以放如归档库。
SPI项目可以做为一个项目进行管理，纳入CM:
 - **EPG**个人编写文档时在工作库中，提交审查及审查后放在受控库中，领导审批后放在基线库中，正式发布后放入规范与标准库中。
- (8) 组织级度量数据库
 - 是将项目的度量数据从项目的配置管理库，从归档库中单独抽取归纳出来，集中在一起而形成的。



库的划分-4





CM的最小文档集

- CM计划
 - 识别出来的CI
 - 识别出来的基线，每条包含哪些CI
 - 权限的分配
 - 库结构的划分（目录结构）
- 识别为CI的准则（CI模板：CI，BL）
- 数据的标识规则（命名）
- CI变更申请表
- 配置审计的检查
- 缺陷跟踪记录（可以PPQA合用一个）
- 配置管理报告
 - 配置项的状态列表（每个CI，当前的版本号，当前状态，）
 - 配置管理的阶段报告（周报，月报，里程碑报告）
- CM的流程定义
 - 包括：CI的变更流程
- 数据备份的制度
- 异地备份



产品与过程质量保证（PPQA）



注意的问题-1

- 检查的对象与时机
 - 所有纳入CI的文档
 - 所有的过程
 - 里程碑评审之前
 - 发布之前
 - 基线建立之前
 - 要明确定义清楚:如除代码和测试用例之外的工作品外100%要检查,代码和测试用例1/4到1/3要检查,关键部位,关键代码必须检查
- 抽检的原则:
 - 代码
 - 测试用例
 - 活动



注意的问题-2

- PPQA的PPQA
 - 互查
 - QA经理检查
 - 外部QA检查
- EPG的PPQA



检查单的编写

- 区分专家与PPQA使用的
- 检查单要根据命中率调整
- 检查项描述要准确,没有二义性
- 对检查项要分类
- 要分角色设计检查单
- 要从形式到本质



PPQA的最小文档集

- PPQA过程定义
- 检查单的模板
- 每个过程
- 每个工作产品
- PPQA的策略
- 检查对象的定义
- 检查时机的定义
- 抽查的原则
- 不符合项记录模板
- 不符合项跟踪流程
- PPQA计划模板
- PPQA报告（汇总报告，周报，月报，阶段报告）
- 按时间分析趋势
- 按PA分析趋势



度量



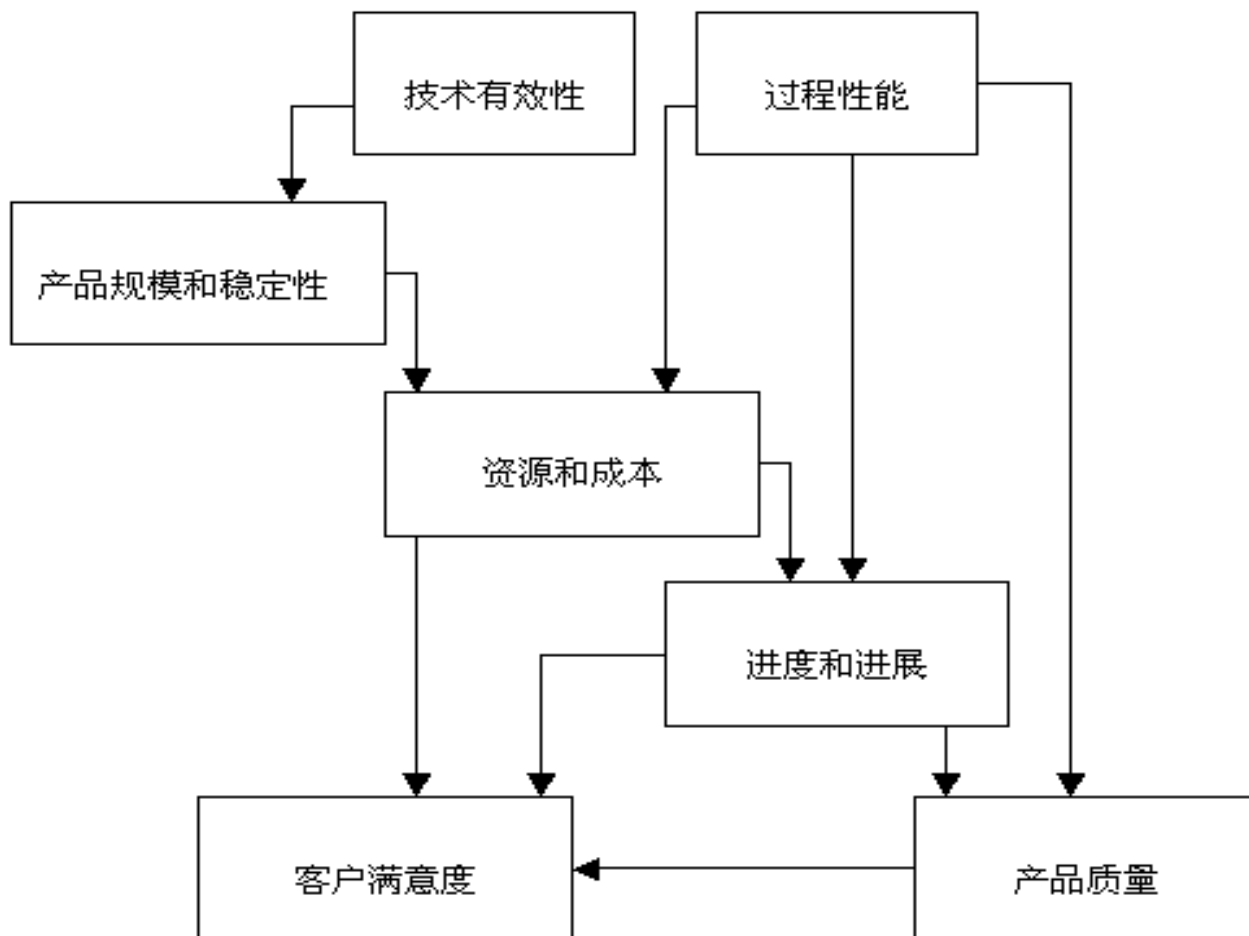
关于度量与分析

- **S1:识别产生度量需求的人**
 - 高层
 - PM
 - 质量管理人员
 - 开发人员
- **S2:访谈相关人员**
 - 第一步根据公司和项目的需要，确定度量目的和度量的信息需要
- **S3:设计信息需求的实现,建立度量的信息模型**
 - 根据度量的信息需要，利用“实用软件度量”的办法，按指示器、派生测量、基本测量、属性进行分解
- **S4:建立度量的采集规程,平衡度量需要与成本**
 - 分解后确定属性的最小集，即利用这些属性就可以得到度量的信息需要
 - 这些属性是在哪些基本表格中填写的。大致有这些基本表格：工作日志表、缺陷日志表、变更日志表等。这些基本表格一定要简洁，尽量少出现重复的信息
 - 通过属性得到的基本测量、派生测量、指示器体现在哪些状态报告中，如项目周报、里程碑报告、项目总结报告、组织级度量库（包括哪些文件）等



五项基本度量

- 进度
- 规模
- 工作量
- 质量
- 过程性能





可以建立的能力基线

- 各测试阶段的测试用例密度
- 各测试阶段的缺陷密度
- 评审工作量占各阶段的比例
- 各测试阶段测试执行工作量与缺陷对应工作量的比率
- 各种语言的纯编码生产率
- 各阶段的生产率
- 主要工种工作量占阶段工作量的比例
- 设计阶段的文档规模（页/KL）
- 文档评审效率
- 代码评审效率
- 测试用例编制效率
- 测试用例评审效率



度量过程最小文档集

- 1 过程定义
- 2 度量的数据模型定义：度量目标，度量元,度量方法 ,数据采集方法,采集时间
- 3 度量结果记录与分析文件
- 4 度量结果记录文件
- 5 检查单
 - 检查数据的准确性



度量元的分类

- 项目级与组织级
 - 组织级:高层+管理人员,阶段性获取,项目组不能裁剪,性能基线
 - 项目级:PM+开发人员,实时获取
- 过程度量与产品度量
 - 过程度量
 - 生产率
 - 质量
 - 资源度量
 - 缺陷注入率
 - 缺陷清除的效率
 - 产品度量
 - 规模
 - 可靠性
 - 质量
 - 代码复杂度
 - 功能



注意问题

- 组织级的活动与项目级的不同



度量元的描述内容

- A 计量单位
- B 采集方法
- C 验证方法
- D 采集周期、时间点
- E 计算规则
- F 优先级
- G 责任分配
- H 度量对象与属性
- I 数据存储位置
- J 对应的指示器
- K 分类
- L 刻度
- M 需求者
- N 决策准则



资源模型概述

- 资源模型是指工作量的花费模型，以用来确定：
 - 软件生命周期各个阶段的时间跨度以及这些时间跨度与总时间跨度之比
 - 软件生命周期各个阶段所花费的工作量以及这些工作量与总工作量之比
 - 软件生命周期某个阶段内不同角色（工种）所花费的工作量以及这些工作量与该阶段所花费的总工作量之比
- 一个高成熟度组织，必须建立自己的项目级和组织级资源模型
 - 注意区分项目的规模、类型和所采用的软件生命周期
 - 关注过程和产品数据的记录、度量、统计、积累和分析
 - 组织级要有定义角色（工种）的文件
 - 特别注意日填工作日志



资源模型的用途

- 资源模型是提高估计准确度的基础
- 资源模型可用来判断过程的质量
 - 资源模型曲线有无断点和跳变
 - 各个比例是否有异常，特别是：
 - 在生命周期各个阶段中同行评审所占的比例
 - 设计在生命周期中所占的比例
 - 可以给出用于过程改进、各类管理、返工、培训和评审的工作量及其所占相应总工作量的比例
- 资源模型是过程改进的重要武器



资源模型应该应用和改进

- 各个项目要参考组织级资源模型中的历史数据，对进度和工作量进行估计（预测），以提高估计（预测）的准确度
- 在建立资源模型时，要注意以下要求：
 - 加大数据点的采集密度（一般以周为单位，所以要强调工作日志的日填、周报、月分析）
 - 加大资源模型的绘制密度（一般以等价周为单位）
 - 要度量过程改进效果，并进行量化的分析，例如：
 - 一段时间内工作效率的提高情况
 - 一段时间内缺陷率的减少情况
 - 一段时间内估计准确度的提高情况
- 应该在使用中改进资源模型：
 - 应随着数据量的增多和准确度的提高而不断完善
 - 在使用时应引入必要的修正系数



资源模型的基本属性

No.	基本特性名称	基本特性描述
1	系统的一般特征	系统名1， ...， 系统名n（至少要8个项目）
		所选用的工作语言（如C++， Perl， Java等）
		项目的成功程度
2	系统类型	紧密结合式， 半结合式， 分离式
		操作系统， 编译系统， 应用软件等
	产品规模	需求或/和代码功能点数， 代码行数， 用例数等
		微， 小， 中， 大， 巨
	生命周期类型	如瀑布模型等
项目组规模	微， 小， 中， 大， 巨	
3	总工作量（单位是人周 或人月）	微， 小， 中， 大， 巨（给出具体数值）
		生命周期各个阶段工作量与总工作量之比
		生命周期各个阶段内不同工种的工作量与该阶段总工作量之比
	项目周期（单位是周 或月）	该项目总的时间跨度
		生命周期各个阶段时间跨度与总的时间跨度之比
4	系统的复杂程度	程序复杂性（MaCabe/HalStead）
		计算复杂性
	采用技术的新颖程度	人员技术水平相对选用的工作语言来说
	人员情况	专业水平， 敬业精神， 士气， 健康状态等



项目级资源模型

- 组织级
 - 要定义生命周期类型
 - 要定义系统类型
 - 要定义产品的物理规模和统计生产率的规模
 - 要定义工种类型
 - 要定义统一的个人日志模板
 - 要求每个项目组成员遵循“日填、周报、月分析”的原则，按项目分别填写个人日志
 - 要求在项目结束时，说明这个项目的成功程度
- 项目级
 - 要根据系统类型、产品规模和生命周期类型三个不同属性，分别建立每个项目的资源模型（统计数据和绘制曲线）
 - 每个项目按周将个人日志中填写的数据汇集
 - 应该建造自动汇集数据的工具



建立项目资源模型的步骤

- 从与项目有关的各工种人员的工作日志或项目周报中，每周累计某个阶段内不同角色（工种）所花费的工作量（人日）填入表1，并计算出各阶段工作量总计、各工种工作量总计和总工作量总计
- 把时间顺序归化为无量纲的时间顺序，即把每个时间点的时间量值除以总工作周期；计算出各工种各阶段工作量与总工作量之比，填入表2，得到工作量按阶段、按工种分布的比例
- 从相应项目的周报、月报或项目总结报告得到各个阶段的时间跨度及项目总花费时间，填入表3，并计算各个阶段的时间跨度与项目总时间之比，即各个阶段花费的时间按阶段分布的比例



某个阶段不同角色所花费的工作量

阶段	需求阶段	设计	编码/单元测试	交付	
表1 软件生命周期某3个阶段内不同角色（工种）所花费的工作量	第1周	第2周	第3周				总计
需求（人日）	15.0	2.0	17.0
设计（人日）	2.0	5.0	32.0
编码（人日）	0.0	2.0	122.0
测试（人日）	0.0	1.0	94.0
PM（人日）	0.0	1.0	5.0
QA（人日）	2.0	2.0	8.0
CM（人日）	0.5	1.0	7.0
MA（人日）	0.5	0.5	6.0
同行评审（人日）	5.0	0.0	11.0
各阶段总计（人日）	27.0	13.5	302.0
	40.5	302.0



工作量按阶段、按工种分布的比例

表2 工作量按阶段、按工种分布的比例（%）

阶段和时间顺序 %	需求阶段		设计阶段		编码和单元测试阶段				集成测试阶段		交付阶段	工作量按工种的 分布比例（%）
	a%	2a%	3a%								100 %	
需求%	0.00	5.20	0.00	0.01	5.12%
设计%	0.00	0.00	5.08	8.21	15.30%
编码%	0.00	0.00	0.00	0.00	38.40%
测试%	0.00	0.05	0.05	0.11	24.02%
PM%	0.00	0.05	0.05	0.11	2.02%
QA%	0.00	0.05	0.04	0.04	4.10%
CM%	0.00	0.01	0.01	0.02	4.02%
MA%	0.00	0.02	0.01	0.02	3.01%
同行评审%	0.00	0.05	0.03	0.04	4.01%
工作量按阶段的 分布比例%	0.00	6.47	5.24	8.45	100.00%
注：现在表中的数据（不是指其已经用红色表示的那些总和值）是相对总工作量来计算的。 表中的数据也可按相对该阶段总工作量的%来计算。												
	8.47%	47.7%	13.69%	13.69%	6.31%	6.31%	6.31%	6.31%	10.23%	10.23%	8.30%	100.00%



各个阶段时间跨度值以及这些值之比

阶段时间跨度	立项阶段	方案阶段	设计阶段	实施阶段	联试阶段	验收阶段	定型阶段	维护阶段	总计
分布比例	0—10	11—20	21—32	33—140	141—148	149—152	153—160	161—200	
表3-1 软件生命周期各个阶段的时间跨度及这些时间跨度之比	4	8	40	200					
时间跨度按阶段的分布比例	5%	5%	6%	54	4%	2%	4%	20%	100%



组织级资源模型

- 将每个项目的资源模型归化为无量纲的只有百分比的等价资源模型
 - 数据采集密度为周，工作量的量纲为人周（或人天）
- 将每个项目按系统类型、产品规模和生命周期类型三个不同属性分别建立的资源模型进行汇总
 - 汇总时数据采集密度为等价周
 - 计算各个里程碑等价值的算术平均值（每个项目的最后一个里程碑值肯定为100%）
 - 计算每两个里程碑之间的各种不同工种的工作量等价值的算术平均值
 - 计算每两个里程碑之间的各种不同工种的工作量等价值之和，其和应该为100%



步骤1：项目工作量、时间跨度按阶段分布表 - 1

- 步骤1：建立每个项目工作量、时间跨度按阶段分布表，如表4.1和表4.2

表4.1 项目1工作量、时间跨度按阶段分布的资源模型

阶 段	立项阶段	方案阶段	设计阶段	实施阶段	联试阶段	验收阶段	定型阶段	维护阶段	总 计
时间跨度(周)	1.0	1.0	1.2	10.8	0.8	0.4	0.8	4.0	20.0
时间跨度按阶段分布比例 (%)	5.00	5.00	6.00	54.00	4.00	2.00	4.00	20.00	100.00
工作量 (人日)	5.0	8.0	31.0	240.0	20.0	11.0	25.0	160.0	500.0
工作量按阶段分布比例 (%)	1.00	1.60	6.20	48.00	4.00	2.20	5.00	32.00	100.00



步骤1：项目工作量、时间跨度按阶段分布表 - 2

表4.2 项目2工作量、时间跨度按阶段分布的资源模型

阶 段	立项阶段	方案阶段	设计阶段	实施阶段	联试阶段	验收阶段	定型阶段	维护阶段	总 计
时间跨度(周)	1.0	1.2	1.1	10.0	1.0	0.4	0.9	4.6	20.2
时间跨度按阶段分布比例 (%)	4.95	5.94	5.45	49.5	4.95	1.98	4.46	22.80	100.00
工作量 (人日)	5.3	7.2	33.0	248.0	19.5	10.0	27.0	163.0	513.0
工作量按阶段分布比例 (%)	1.03	1.40	6.43	48.40	3.80	1.95	5.30	31.77	100.00



步骤2：组织级工作量、时间跨度按阶段分布表

- 步骤2：在系统类型、产品规模和生命周期类型三个属性接近的N个“同类项目”的资源模型基础上，建立组织级的工作量、时间跨度按阶段分布表，例如表4.3
- 在此例中用了上述两个项目的数据，即N=2，通常应在8个“同类项目”的基础上，建立组织级的资源模型
- 表4.3中
 - 时间跨度(周) $T_i = \sum t_i / N$
 - 时间跨度之比 (%) = $(T_i / \sum T_i) \times 100\%$
 - 工作量(人日) $F_i = \sum f_i / N$
 - 工作量与总工作量之比 (%) = $(F_i / \sum F_i) \times 100\%$



步骤2：组织级工作量、时间跨度按阶段分布表

4.3：组织级的工作量、时间跨度按阶段分布的资源模型

阶段	立项阶段	方案阶段	设计阶段	实施阶段	联试阶段	验收阶段	定型阶段	维护阶段	总计 Σ
时间跨度(周) T_i	1.0	1.1	1.15	10.4	0.9	0.4	0.85	4.3	20.1
时间跨度之比 (%)	4.98	5.47	5.72	51.74	4.50	1.91	4.23	21.40	100.0
工作量 (人日)	5.15	7.6	32	244	19.75	10.5	26	161.5	506.5
各阶段工作量与总工作量之比 (%)	1.01	1.50	6.32	48.17	3.90	2.07	5.13	31.9	100



构造组织级工作量按工种分布表的步骤 - 1

- 步骤1：建立每个项目工作量（人日）按工种分布表，如表5
- 步骤2：在系统类型、产品规模和生命周期类型三个属性接近的N个“同类项目”的资源模型基础上，建立组织级工作量（人日）的按工种分布表

组织级各工种工作量平均值： F_i （人日）= $\sum \text{该工种 } f_i / N$

N个“同类项目”

组织级工作量按工种分布比例： P_i （%）= $F_i / \sum F_i$

在例子表5中为简单起见，只列入两个项目，即N=2



构造组织级工作量按工种分布表的步骤 - 2

表5 组织级工作量按工种分布的资源模型

	计划	需求	设计	编码	测试	QA	CM	MA	评审	总计
项目一工作量 f_i	5	17	32	122	94	8	7	6	11	302
项目二工作量 f_i	7	14	35	131	93	10	6	6	13	315
组织级各工种 工作量平均值 F_i (人日)	6.0	15.5	33.5	131.5	93.5	9.0	6.5	6.0	12.0	313.5
组织级工作量 按工种分布比例 P_i (%)	1.91	4.92	10.69	41.95	29.83	2.87	2.07	1.92	3.83	100.00



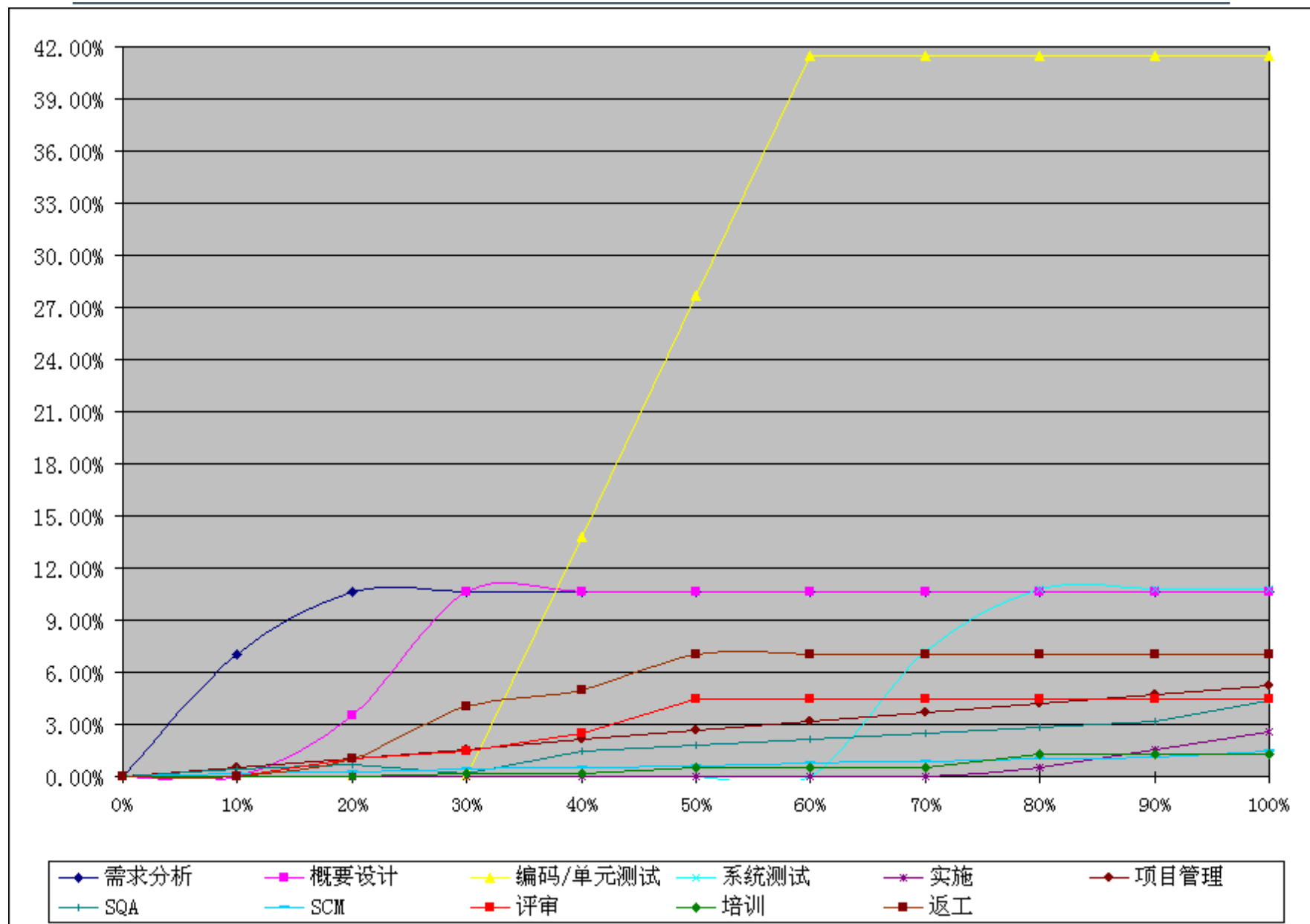
资源模型数据示例

	工作总量 %	需求分析 %	概要设计 %	编码/ 单元测试 %	系统测试 %	实施 %	项目管理 %	评审 %	SQA %	SCM %	培训 %	返工 %
		10.60	10.60	41.50	10.80	2.60	5.20	4.50	4.40	1.50	1.30	7.00
0%	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10%	8.14	7.07	0.00	0.00	0.00	0.00	0.52	0.00	0.40	0.15	0.00	0.00
20%	18.14	10.60	3.53	0.00	0.00	0.00	1.04	1.00	0.71	0.26	0.00	1.00
30%	29.08	10.60	10.60	0.00	0.00	0.00	1.58	1.50	0.20	0.40	0.20	4.00
40%	46.78	10.60	10.60	13.83	0.00	0.00	2.11	2.50	1.43	0.51	0.20	5.00
50%	65.94	10.60	10.60	27.67	0.00	0.00	2.64	4.50	1.79	0.64	0.50	7.00
60%	80.77	10.60	10.60	41.50	0.00	0.00	3.16	4.50	2.14	0.77	0.50	7.00
70%	88.99	10.60	10.60	41.50	7.20	0.00	3.69	4.50	2.50	0.90	0.50	7.00
80%	94.92	10.60	10.60	41.50	10.80	0.52	4.22	4.50	2.86	1.02	1.30	7.00
90%	96.96	10.60	10.60	41.50	10.80	1.56	4.74	4.50	3.21	1.15	1.30	7.00
100%	100.00	10.60	10.60	41.50	10.80	2.60	5.20	4.50	4.40	1.50	1.30	7.00

根据你们的经验，这些资源模型数据有没有问题？

有哪些问题？

为什么？





资源模型数据示例

t%	工作总量 %	需求分析 %	概要设计 %	编码/ 单元测试 %	系统测试 %	实施 %	项目管理 %	评审 %	SQA %	SCM %	培训 %	返工 %
0%	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10%	8.14	7.07	0.00	0.00	0.00	0.00	0.52	0.00	0.40	0.15	0.00	0.00
20%	18.14	10.60	3.53	0.00	0.00	0.00	1.04	1.00	0.71	0.26	0.00	1.00
30%	29.08	10.60	10.60	0.00	0.00	0.00	1.58	1.50	0.20	0.40	0.20	4.00
40%	46.78	10.60	10.60	13.83	0.00	0.00	2.11	2.50	1.43	0.51	0.20	5.00
50%	65.94	10.60	10.60	27.67	0.00	0.00	2.64	4.50	1.79	0.64	0.50	7.00
60%	80.77	10.60	10.60	41.50	0.00	0.00	3.16	4.50	2.14	0.77	0.50	7.00
70%	88.99	10.60	10.60	41.50	7.20	0.00	3.69	4.50	2.50	0.90	0.50	7.00
80%	94.92	10.60	10.60	41.50	10.80	0.52	4.22	4.50	2.86	1.02	1.30	7.00
90%	96.96	10.60	10.60	41.50	10.80	1.56	4.74	4.50	3.21	1.15	1.30	7.00
100%	100.00	10.60	10.60	41.50	10.80	2.60	5.20	4.50	4.40	1.50	1.30	7.00



决策支持与分析 (DAR)



最小文档集

- (1)DAR过程定义 DAR sp1.4
- (2)项目决策指南 DAR sp1.1
 - 对组织级或项目组中哪些类型或哪些具体的决策行为必须执行DAR过程。这个指南一般是一个组织级的指南，即使定义为项目级的指南
- (3)项目决策事项列表 DAR sp1.1
 - 在某个项目内哪些具体问题需要执行哪个正式决策程序？
 - 正式决策程序的可能有多个，对不同类的问题，严重程度不同，决策程序可能不同。
- (4)决策分析与解决方案报告 DAR sp1.2, DAR sp1.3, DAR sp1.4, DAR sp1.5 , DAR sp1.6
 - 决策指标要量化
 - 注意：对于不同的决策，指标不同，列一个模版就可以了。



决策指南的制定

- 1指导原则：用于确定在什么情况下要求结构化决策的典型指导原则包括：（模型中的描述）
 - 决策直接关系到的问题可能是中风险或高风险的
 - 决策关系到对配置管理之下的工作产品的变更
 - 决策不当可能造成进度拖延
 - 决策影响到实现项目目标的能力
 - 与决策的影响相比较，决策过程的成本可以接受
- 2对需决策的问题进行分类：（这是我们建议的）
 - （1）目标性决策：
 - 是自己开发？还是买成品？还是外包给其他公司？还是联合开发？
 - 项目立项的决策
 - 是否继续在某个技术上投入？如：对技术平台的投入？
 - 给客户先实施1.0，将来再升级到2.0，还是让客户等一下，直接实施2.0 的软件？
 - （2）对需求影响比较大的：
 - 需求的重大变更，变更影响的工作量有多大时。



决策指南的制定

- (3) 对进度影响比较大的
- (4) 对质量影响比较大的
- (5) 对技术影响比较大的
 - 在软件开发过程中是否采用持久对象层的技术?
 - 是采用哪种技术路线? 采用JAVA还是.net路线?
 - 当前的技术路线与未来的技术路线是不一致的, 如何权衡这些问题?
 - 数据库系统的选择
- (6) 对工具影响比较大的
- (7) 对资源影响比较大的
 - 是采用外购的报表构件还是自己开发?
- (8) 对环境影响比较大的
 - 是否要进行封闭开发?
 - 是否在现场进行软件开发?
- 3 不同类型的决策, 正规决策的流程可能不同 (这在模型中有描述)
- 流程不同
- 参与决策的人员的数量
- 参与决策的人员级别等也可能不同



在模型中明确提到的需要执行DAR的PA

过程域	实践
SAM	SP 1.2 Select suppliers
TS	SP 1.1 Develop Detailed Alternative Solutions and Selection Criteria SP 1.3 Select Product-Component Solutions SP 2.3 Design Interfaces Using Criteria SP 2.4 Perform Make, Buy, or Reuse Analyses
PI	SP 1.1 Determine Integration Sequence
OT	SP 1.4 Establish Training Capability
IPM	SP 4.1 Determine Integrated Team Structure for the Project
RSK	在概述中提到了



正规评价方法的要点

- 要有不同的方案进行比较,至少有2个方案
 - 可以一个方案比较完整,一个比较简单
- 应该有个判断方案好坏的准则
- 不同类型的软件准则不同
- 评价的方法
 - 评审
 - 测试
 - 模拟
 - 情景分析



项目策划、项目监督与跟踪



计划的分类与制订时机

- 项目计划
 - 主计划
 - 详细计划
- 三个时机
 - 立项
 - SRS
 - 变更



项目计划的2个层次-1

- 项目计划一般根据项目的周期,分成2个层次或3个层次,即阶段计划(子阶段计划)、详细计划(进度表):
 - 阶段计划,也就是里程碑计划
 - 该计划没有明细到每个任务,而是比较粗的计划
 - 该计划代表了项目组对外的承诺
 - 阶段计划的颗粒度一般是以月为单位,该计划一般不容易变化,除非在里程碑评审时发现重大偏差
 - 一般如果某个阶段比较长,比如2个月,则需要有这个阶段的“子阶段计划”



项目计划的2个层次-2

- 每个阶段的详细计划，在上一个里程碑结束时要细化
 - 设计阶段的详细计划是在需求分析阶段结束时细化，当然也可以在设计阶段的初期做。一般在做上一个阶段的里程碑评审时，要对下一个阶段的详细计划进行评审，当然不是具体到每个任务几天的工作量，也是对设计阶段的“阶段计划”进行评审
 - 对于该计划要求任务的颗粒度不要大于5天。一般的公司都是定义为2天或者3天。
 - 项目的详细的计划一般是一般每半月要调整一次。
- 阶段计划一定要入基线库。项目的进度计划可以不入基线库。详细计划放入受控库，详细计划的调整也要通知相关人员。这样比较灵活也控制了进度风险



日志

- 进展日志
 - 谁?
 - 什么时间?
 - 做了什么事情?
 - 事情的分类?
 - 工作量?
 - 工作产品的规模?
- 质量日志
 - 谁?
 - 什么时间?
 - 什么位置?(模块等)
 - 犯了什么错误?
 - 缺陷描述
 - 类型
 - 严重程度



确保日志的质量

- 日志的质量要求
 - 及时性
 - 每天都要记录
 - 真实性
 - 不与个人考核挂钩
 - 可统计
 - 任务要有标准分类
- 建立日志的抽查机制
 - 每个人的日志在什么时间区间内必须抽查一次



如何开项目进展周例会？

- 会前准备(1到2个小时)::
 - 项目经理检查每个人工作
 - 收集并解决问题
 - 安排每个人下周的任务
- 会议中(控制在半小时以内):
 - 项目经理通报上周进展 (2分钟)
 - 上周存在的共性问题(5分钟)
 - 经验教训的总结(5分钟)
 - 下周项目组的总体计划 (5分钟)
 - 其他问题(5分钟)
- 会议后:
 - 跟踪问题的关闭
 - 会议纪要
 - 不要超过1页纸
 - 记录结论和经验教训
 - 分发到每个人



进展周报的内容

- 在进展周报中，按如下的顺序排列资料：
 - 上周任务完成概况（文字描述）
 - 上周工作存在的问题（文字描述）
 - 下周的任务安排（文字描述）
 - 上周的经验教训总结（文字描述）
 - 然后列出各种统计数据和表格
 - 甲乙双方确认
- 这样更能抓住重点，更实用



项目管理

- 任务依赖关系：
 - 项目组内部的依赖： 日程表中
 - 项目组外部的依赖： 日程表中
 - 如何跟踪和控制外部依赖？
- 例会：无论大小都得开会。什么人参加会议，根据项目组团队规模？分层次
- 项目组有哪些数据要提交到组织级？



反应式控制模式

- 在CMM/CMMI L2，控制是反应式的。反应式控制模式的要求是：
 - 每个任务都要规定期望值
 - 每个任务都要规定该期望值的控制阈值
 - 要根据商业目标设置控制阈值
 - 区分关键路径和非关键路径
 - 按相对偏差进行控制，按相对偏差和绝对偏差进行报告
 - 设置必要的提前量（系统动力学问题），区分控制阈值和预警阈值



预测式控制模式

- 在CMM/CMMI L3，大部分控制应该是预测式的；而在CMM/CMMI L4&L5，所有控制都应该是预测式的
- 预测式控制模式应该在反应式控制模式的基础上，再考虑：
 - 将非关键路径上的松弛时间等比例地、动态地进行定量分配
 - 非关键路径的控制阈值可以定义为：在原有关键路径的控制阈值的基础上，再考虑将非关键路径的松弛天数，按照该非关键路径上任务的周期进行等比例分配
 - 应用外推法计算出趋势，并用图形来表示。在分析时，要根据当前点及其前面的**2-4**个点的数据进行非线性外推，以确定预警阈值是否超出了预定的范围



关键路径主动式跟踪

- (1) 通过任务依赖关系，识别关键路径
- (2) 关键路径的控制阈值、警戒阈值
- (3) 在进展报告中，要报告出关键任务的相对偏离和绝对偏离
- (4) 关注即将转化成关键路径的非关键路径
- (5) 关注与关键路径上相关联的非关键任务



数据管理与配置管理

- 数据管理的内容
 - 识别数据: 要管理哪些数据?
 - 内容定义: 应该包含哪些章节, 每个章节应该如何写。
 - 形式定义: 是WORD, 还是PPT, 是表还是文字等
 - 权限管理: 谁有权力读取?
- 配置管理的内容
 - 识别CI
 - 定义基线
 - 管理CI的变更
 - CI的状态报告
 - 基线的审计
- 对这些数据能识别出来具体的文档的, 就识别出来具体的文档的, 就识别出类别来, 然后按类来定义他们的属性, 对他们进行管理。比如, 无法识别出来将来会发生那些邮件的往来, 但是, 按类来管理, 这种管理是粗放式管理





数据管理实践

- 识别出所有需要管理的数据
- 定义清楚各种数据的来源,形式,存储介质
- 定义清楚每种数据应描述的内容与采用的格式
- 定义清楚数据存储的目录结构
- 定义清楚数据的存取权限,分发范围与分发方式
- 数据采集的进度安排



PP文档的最小集-1

- WBS的分解指南 PP SP1.1
 - 如何做WBS分解?
 - 在WBS分解中最容易遗漏的任务有哪些?
 - 每个PA GP2.2
- 规模估计及跟踪记录模板 PP SP1.2 PMC SP1.1
- 规模估计的指南 PP SP1.2
 - 如何做规模估计?
 - DELPHI方法
 - PERT方法
 - FFP方法
 - 其他方法
- 组织级的生命周期模型定义 PP SP1.3
- 工作量与成本估计的指南（规程） PP SP1.4
 - 如何从规模转化为工作量?
 - 如何估计成本?
- 工作量与成本（预算）的估计记录 ‘ PP SP1.4 PMC SP1.1



PP文档的最小集-2

- 项目计划书 PP SP2.1,PP SP2.7
- 数据管理表 PP SP2.3,PMC SP1.4
- 穷举出所有的可能数据
- 项目资源列表PP SP2.4,PMC SP1.1
- 知识技能表 PP SP2.5,PMC SP1.1
- 人员职责定义表PP SP2.6,PMC SP1.5
- 评审记录(公司统一,所有的评审都采用同一种记录格式)
- 缺陷跟踪表(公司统一)
- PP过程定义
 - 无SRS
 - 有SRS
 - 计划变更



PMC 的最小文档集

- 日志模板 PMC SP1.6
- 周例会的规程 PMC SP1.6
- 周进展报告 PMC SP1.6
- 里程碑评审的规程 PMC SP1.7
- 里程碑评审报告模板 PMC SP1.7
- 项目的总结报告模板 PMC SP1.7, IPM SP 1.5
- PMC 流程定义



预测式控制

- 根据前3周的值预测后一周的趋势
 - $Y=AX+B$
 - 已知:
 - 第1周的进度偏差率 -3.45%
 - 第2周的进度偏差率 -27.59%
 - 第3周的进度偏差率 -3.45%
 - 预测第4周的进度偏差率 -11%
 - 在EXCEL中使用trend(前3周的值,{0;1;2},{3})函数,线形回归预测第4周
- 可以预测的对象
 - 进度偏差率
 - 工作量偏差率
- 当发现预测的偏差超过控制阈值,需要文字说明,分析原因,并采取纠正措施



历史项目数据的参考

- 个人生产率
- 整体生产率
- 估计偏差率
- 类似模块的规模



供应商合同管理 (SAM)



SAM的过程定义

- 适用范围:软件、硬件的采购、开源代码等
- 活动:
 - 确定产品和服务清单、类型
 - 类型: 默认协议、正式协议 (从规范不考虑招标)
 - 发起人提出采购申请, 审批采购申请 (遵循公司现有的购物流程)
 - 在采购申请中, 要确定产品和服务清单、类型、用途、预算、到货日期
 - 制订SAM计划
 - 选择供应商
 - 建立候选供应商清单 (隐含有一个合格供应商清单)
 - 确定供应商的选择准则(量化)
 - 按准则评价和选择供应商 (要说明经领导审批)
 - 如果能得到现货的产品, 要考虑对现货产品做测试或评审
 - 如果不能得到现货的产品, 要获取相关的资源并进行评估
 - 签署正式协议, 并得到高层领导的审批
 - 执行并监督供应商协议 (遵循公司现有的财务流程)
 - 验收供应商交付的产品
 - 运用供应商移交的产品



SAM

- 模板
- 采购单
- SAM计划
- 供应商评价报告：候选供应商、准则、评价结果等，要有领导审批的意见
- 正式协议（不出模板）
- 协议执行情况（包含验收结果，移交情况，培训等）



组织过程焦点 (OPF)



组织过程焦点（OPF） -1

- 在OPF中提到的和过程改进的计划包括了如下几种：
 - 组织过程改进计划
 - 评估计划
 - 过程行动计划
 - 试点计划
 - （部署）推广计划
- 其中，组织过程改进计划是后四种计划的集合和概要
- 过程改进行动计划是针对每次评估（差距分析，小型评估，预评估，正式评估）后针对每个弱项来制定行动项、责任人、修改时间等
- 过程行动计划是详细的实施计划。与组织过程改进计划不同，过程行动计划针对的是具体改进，这些改进是由评估所发现的薄弱环节和实施人员的反馈意见和合理化建议



组织过程焦点（OPF）-2

- 在模型中，对过程改进行动计划中应包括的内容是这样描述的：
 - 过程改进基础设施
 - 过程改进目的
 - 将要处理的各项改进
 - 用于策划和跟踪过程行动的规程
 - 关于试行和实施过程行动的策略
 - 关于实施过程行动的责任和权限
 - 关于实施过程行动的资源、进度和相应的分配
 - 用于确定过程行动效果的方法
 - 与过程行动计划相关联的风险
- 在过程行动计划中包含的活动如：
 - 分配弱项的责任人
 - 制定改进措施
 - 起草过程文件
 - 购买过程改进的工具
 - 选择试点项目
 - 评审过程文件
 - 等等



组织过程焦点（OPF）-3

- 对于确定待改进项的方法，模型是这样描述的：
 - 对过程进行度量，并分析度量结果
 - 评审过程的有效性以及适用性
 - 评审通过剪裁OSP（组织标准过程）得到的经验教训
 - 评审在过程实施中取得的经验教训
 - 评审管理者和员工以及其他相关人员提出的过程改进建议
 - 从高层经理和领导那儿得到的过程改进输入
 - 检查过程改进和其他过程相关评审的结果
 - 评审其他组织过程改进创议的结果



组织过程焦点（OPF）-4

- 试点计划是指针对某项改进，选择好试点项目或试点部门，在试点项目中的使用计划。试点项目的选择要有一定的技巧。比如：
 - 不要选择可控程度很低的项目，也不要选择太容易管理的项目
 - 前者在实施过程中会出现许多不可控因素，对执行者产生很大的压力，实施得不好，会给后期的全面推广实施带来负面影响
 - 后者虽然顺利实施，但不能将组织级文档存在的问题充分暴露，不利于文档的完善。赛柏科技咨询师将结合自己的经验协助四川长虹技术中心选择试点项目
- 部署也可译为推广，在做好试点之后，如何推广到每个项目中去，包括培训、裁剪、修改等活动。在部署计划中应该包括的内容：
 - 对定义好的过程体系进行培训
 - 识别出需要的裁剪活动
 - 识别出需要的工具
 - 定义进度表
 - 等等



计划的合并问题

- 部署计划和试点计划可以是整体的过程改进计划中的一个章节，也可以是单独的计划。
- 一般来说，**3级**的公司会将刚刚建立的过程体系部署计划和试点计划写在过程改进计划中，而**5级**的公司可能会针对某个改进单独做部署计划和试点计划。也就是说对于**3级**的公司在启动的时候可以将上述的**4**个小计划都写在一个过程改进计划中，当执行过程中，需要对各个部分进行细化，可以针对具体的改进措施出详细的计划。



组织过程焦点（OPF）-5

- 对于合理化建议的管理，模型中是这样描述的：
 - 征求过程改进建议
 - 汇总、归纳过程改进建议
 - 评审过程改进建议
 - 选择要实施的过程改进建议
 - 跟踪过程改进建议实施情况



其他

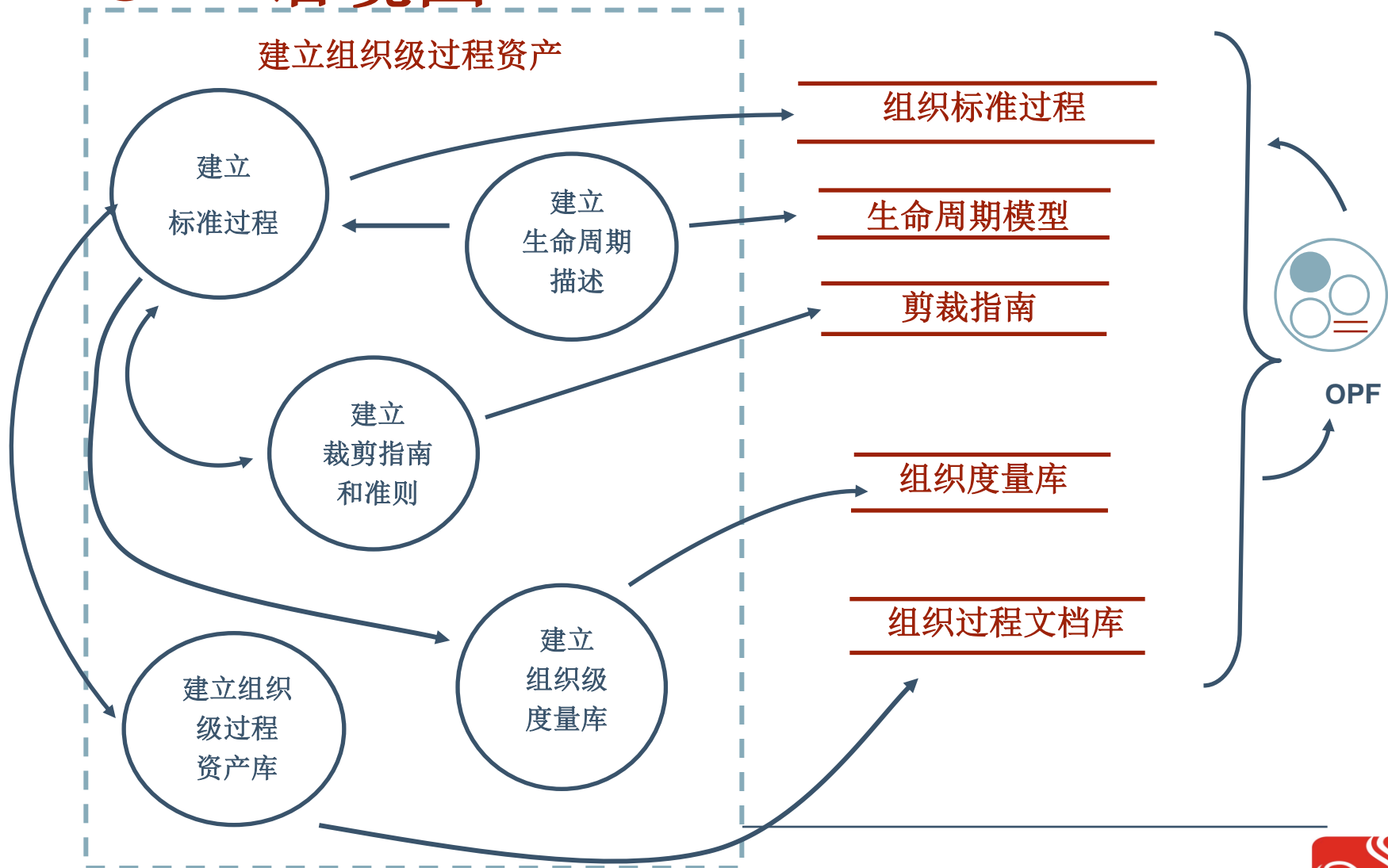
- 建立定期沟通的制度



组织过程定义(OPD)

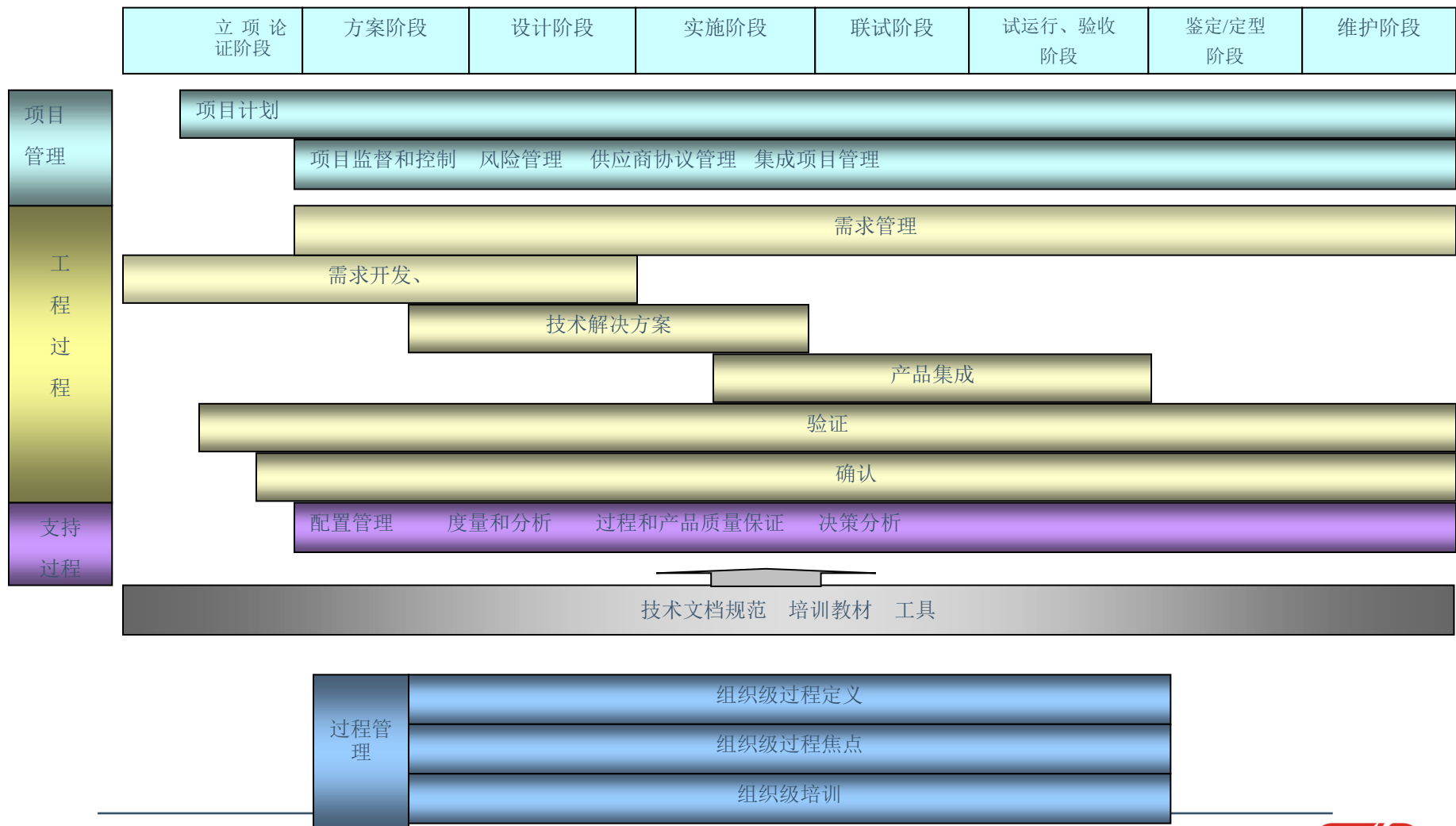


OPD语境图





生命周期阶段与过程之间的关系





2.2 关于定义生命周期模型的若干问题

在哪一个文档中，要明确规定基线工作产品（配置管理计划/项目开发计划，软件生命周期定义，定量管理计划）？

- 在生命周期各个阶段的输出产品定义中，是否应该包括需求跟踪矩阵、测试用例和各种计划？为什么？
- 在需求阶段是否应该开始编写测试用例？
- 在裁剪指南中是否可以规定将设计和编码工作合并为一个阶段？
- 设计阶段的时间跨度占项目总周期的百分比一般应该等于多少？
- 在生命周期定义文档中，同行评审工作量占项目总工作量的百分比一般应该等于多少？
- 在制定开发计划时，应该将WBS中任务的工期定义为几天比较合适（3—5天，6-10天，10-20天）？为什么？



生命周期的裁剪和选择

- 生命周期的裁剪
 - 阶段裁剪
 - 过程裁剪
 - 文档裁剪
- 生命周期的选择
 - 开发方法
 - 项目的特征
 - 开发组织的特征



生命周期的选择

- 生命周期是过程改进进程中的一个主文档，各个组织要根据各自的情况：
 - 制订适用的生命周期
 - 制订适用的生命周期裁剪指南
- 需要定义哪些类型的生命周期？主要的准则是：
 - 前**2-3**年中证明行之有效的
 - 预测在最近**2-3**年中最有生命力的
 - 注意到瀑布模型是各类生命周期的内核
- 在定义生命周期时，要注意规定同行评审的力度



需求阶段要考虑的问题

- 明确性
- 完整性
- 易理解性
- 稳定性
- 需求变化率（对新增的、删除的和修改的需求数分别进行度量，并计算其和）
- 需求提出变更的生命周期阶段，同一需求的变化次数，并按逆序排序
- 可追溯性：双向可追溯性（用户需求与系统需求不能相互对应的数量，用户需求与编码和测试不能相互对应的数量等）
- 易测试性



需求阶段出口准则示例

- 已建立了原始需求与需求规格说明之间、需求规格说明与已编写的系统测试用例之间、需求规格说明与设计之间、设计与集成测试用例之间、设计与代码之间以及代码与单元测试之间等6个映射关系（需求跟踪矩阵）
- 每一个软件需求都可编写出相应的测试用例
- 已编写了25-30%的系统测试用例并通过了评审
- 根据组织能力基线和项目最终质量目标制定需求阶段应发现的缺陷数
- 100%修复了需求阶段发现的缺陷
- 已制订了系统测试计划
- 项目开发计划等得到批准
- 需求阶段评审的工作量大于需求阶段总工作量的1/4
- 客户、高层经理、项目经理及项目相关人员参加了需求的评审

需求阶段评审的功能数占需求阶段已确定的功能数的100%



设计阶段要考虑的问题

- 在确定构架时，对结构的稳定性、柔性、逻辑复杂度和程序结构的形态等是如何考虑的？
- 设计的功能度是否覆盖了所有的需求（功能、性能和接口需求）？是否考虑了潜在需求？
- 在确定设计方案时，是否对构件的选择（购买、复用、委托开发和自行开发）进行了充分的权衡？
- 在确定设计方案时，是否对先进性与可行性进行了充分的权衡（例如，90%的构件应该是经过考验的原则）？
- 是否建立了设计与集成测试用例的映射关系？是否对系统测试用例进行补充和完善？
- 集成测试用例是否覆盖了所有功能（或所有结构）？对集成测试用例是否进行了评审？
- 是否考虑了每阶段要编写的工作文档的数量和质量（内部文档）？
- 是否考虑了每阶段要编写的交付文档的数量和质量（外部文档）？
-



设计阶段出口准则示例

- 设计全部覆盖了需求，并且已经文档化
 - 程序的形态要好
 - 已编写了集成测试用例，并建立了设计和代码之间以及设计和集成测试用例之间的映射关系
 - **SRS**、系统测试计划和测试用例得到评审和批准，并置于**CM**之下
 - 系统技术设计说明通过评审，并置于**CM**之下
 - 全部设计文档必须进行同行评审
 - 评审工作量不低于设计总工作量的1/4
 - 对集成测试用例进行了技术评审（同行评审）
 - 设计阶段发现的开发缺陷小于1.0-2.0/页
 - 设计阶段缺陷清除率大于99.6%
-
- 未清除缺陷中无一级缺陷和二级缺陷



编码和单元测试阶段出口准则示例

- 代码全部覆盖设计
 - goto的使用符合结构化原理
 - 建立了代码和单元测试用例之间以及代码和设计之间的映射关系
 - 成功执行所有单元测试计划中的测试用例
 - 单元测试要求100%语句覆盖
 - 新编代码和核心代码必须经代码评审
 - 同行评审的代码行数不低于25%-35%
 - 评审工作量不低于编码总工作量的35%
 - 编码阶段发现的缺陷应少于3.0/KLOC
-
- 阶段未清除率（逸出率）小于0.2%-0.5/KLOC
 - 未清除缺陷中无一级缺陷。一级缺陷应少于20%



测试出口准则示例

- 完成单元测试，覆盖率可以参考以下要求
 - 语句覆盖率应达到100%
 - 分支覆盖率应达到60%-80%
- 完成集成测试，完成准则可以参考以下要求
 - 100%的集成单元覆盖率
 - 100%的单元调用覆盖率
- 完成配置项测试，完成准则可以参考以下要求
 - 每个软件功能必须至少被1个测试用例覆盖
 - 没有1、2级错误
 - 每500句源代码不多于1个3级错误
 - 每200句源代码不多于2个4、5级错误



裁剪原则

- 里程碑裁剪原则
 - 系统项目和软件项目对里程碑划分的要求不同，如果只是软件项目可以裁剪系统硬件平台部分内容
 - 系统项目的用户要求不同，也可以对里程碑裁剪，如分系统技术设计可以归入系统总体技术设计
- 基线：产品生命周期的基线一般有：
 - 功能基线：系统需求分析阶段的结果，产品为系统总体方案
 - 分配基线：设计阶段需求分析，产品为软件需求规格说明
 - 技术基线：设计阶段的结果，如概要设计或详细设计文档
 - 产品基线：验收阶段的结果，产品为源代码、用户手册、安装手册等



组织过程资产库

- 组织过程资产库中存储的项包括：
 - 组织的方针
 - 已定义过程的描述
 - 规程（例如，进行估计时使用的规程）
 - 开发计划
 - 质量保证计划
 - 培训材料
 - 过程辅助材料（例如，检查表）
 - 经验教训报告
- 注意:可复用构件库并没有要求一定包括在里面



项目实施文档如何纳入组织过程资产库

- 并非所有的项目实施文档都纳入组织过程资产库
- 在模型的2，3级中，有3个实践涉及到将项目组的有价值的文档纳入到组织过程资产中，他们分别是：
 - IPM SP1.5: 侧重于将项目组的经验教训、估计数据等纳入组织过程资产库。
 - OPD SP1.5: 侧重于对组织过程资产库的建立与管理
 - OPF SP2.4: 侧重于把过程改进的经验教训等纳入库中。
- 在OPD 的SP1.5中特别定义了一个子实践2: Specify the criteria for including items in the library. The items are selected based primarily on their relationship to the organization's set of standard processes.
- 到底企业需要将哪些项目组产生的文档放入过程资产库中，需要建立一个准则。



需要建立的过程描述

- 过程体系建立的流程
- 过程体系更新的流程
- 组织级度量元建立与更新的流程
- 组织级度量数据筛选、分析与发布的流程
- 项目过程资产入库的流程
- 最佳实践入库流程
- 会议流程
- 技术交流例会制度
- 管理交流(PPQA,CM,PM)例会制度



- 各个单位所开发的项目规模、用户要求都可能存在较大差异，因此，项目负责人应该根据项目的具体要求，对组织级的产品生命周期文档进行适当的裁剪
- 阶段裁剪原则
 - 项目开发的阶段，除定型阶段可以裁剪外，其它阶段不能裁剪
- 过程裁剪原则
 - 实现过程
 - 立项论证阶段的内容可以裁剪
 - 软件概要设计和详细设计可以合并
 - 其它实现过程不能裁剪
 - 项目管理过程
 - 立项管理过程除合同管理不能裁剪外，其它活动可以裁剪
 - 项目计划、需求管理、项目监督控制等管理过程不能裁剪
 - 支持过程
 - 项目支持过程不能裁剪
- 文档裁剪原则
 - 《软件概要设计说明》和《详细设计说明》文档可以根据情况合并
 - 里程碑中规定主要技术文档原则上不能裁剪



组织级培训（OT）



OT讨论 -1

- 战略性培训需求、岗位培训需求：时间跨度为2年以上
- 培训需求调查：年底进行。半年调查一次
- 年度培训计划：
 - 年初制订，每半年更新一次
 - 要年度计划的跟踪机制和更新机制
- 执行计划：是一个总表，滚动更新
- 每次培训之前，通过电子邮件提前发培训通知到每个学员。在培训通知中，包含对该次培训的课程描述、日程安排、培训地点等
- 讲师管理：
- 培训教材：



OT讨论 -2

- 培训免除：规范中明确免培的机制
- 实施培训
- 学员反馈表、综合评价结果
- 讲师评价表
- 培训实施记录表
- 培训效果调查表



4321法

- 4个方面
 - 岗位职责
 - 知识技能
 - 课程
 - 需要培训的课程:
- 3个矩阵
 - 岗位与知识技能的关系
 - 知识技能与课程的关系
 - 课程与需要培训的课程的关系
- 2个检索
 - 公司中需要培训的课程门数有多少,每门课需要培训的有多少人
 - 每个人需要需要参加培训的课程有哪些
 - 排序,先安排参加人数多的,再安排紧急的.
- 1个培训计划:
 - 根据上面的检索结果,开会征求的意见与个人的愿望,领导意图,个人的职业规划



生命周期模型的选择



生命周期过程描述的12个方面

- 过程角色（Process roles），哪些角色参与本过程的哪些活动，可以用角色-职责矩阵表示
- 适用的过程和产品标准（Applicable process and product standards），包括企业内的或者企业外的（标准没有，客户的标准、行业的标准等没有，放在项目定义过程中）
- 适用的规程、方法、工具和资源（Applicable procedures, methods, tools, and resources）。资源中包括了关键的设备（资源只列各种模板等，现在体现不出）
- 过程性能目的（Process performance objectives）。可用一些量化数据来表示，如周期、生产率和缺陷排除率等
- 入口准则（Entry criteria）
- 输入（Inputs）：哪些文档是该过程或活动的输入
- 活动
- 要收集和使用的产品和过程度量（Product and process measures to be collected and used）（在度量规范中说明，但说明属于哪个过程规范）
- 验证点（如同行评审）（Verification points (e.g., peer reviews)）(过程流图中)
- 输出（Outputs）：输出那些文档，要注意这些输出是否覆盖了模型的要求
- 接口（Interfaces）：与其他过程或规程的衔接关系（总体的过程体系结构，相互之间的关系）(体现在过程活动中)
- 出口准则（Exit criteria）：定义了过程或活动应达到什么要求才算结束了



其他问题



关于文档 -1

- 文档化的目的是：
 - 沟通
 - 存档
 - 复用
- 原则：
 - 最基本的交流文档
 - 人多的时候不可以省略文档
 - 人不熟悉的时候不可以省略文档
 - 产品不可以缺少文档
 - 承诺是不可以省略的
 - 将来要复用的内容必须写文档
 - 客户要求的文档必须写
 - 开发周期比较长



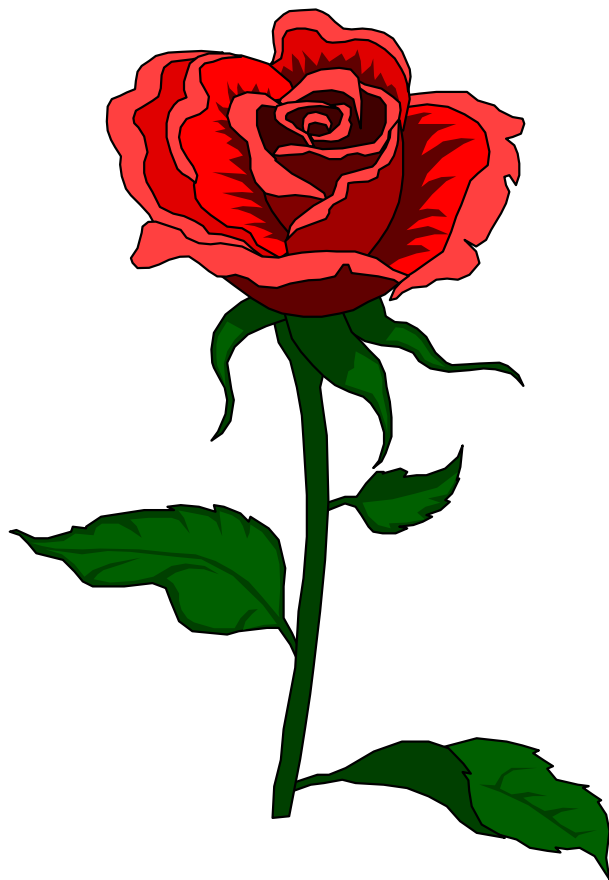
关于文档 -2

- 文档的分类:
 - 工程文档
 - 管理文档
 - 组织过程文档
 - 支持过程文档
- 文档的受众:
 - 全体人员
 - 本项目组的开发人员
 - 管理人员



问题与回答





谢谢



Carnegie Mellon
Software Engineering Institute

义杰 (Jacky)

Lead Consultant

Shanghai Share-Win (上海解元科技)

Shanghai office:

Tel: (86)21-51085518 Fax: (86)21-64853278

Mobile: 18916292126

Email: jacky.yi@share-winning.com.cn

MSN:jacky51888@msn.com

<http://www.sharewin.cn>

