

软件项目管理高级研讨班：补充材料

廖彬山

美国CMU/SEI授权的CMMI主任评估师

Email : liaobs@263.net

手机 : 13910021839

课程安排

- 第一天：项目策划及估算
 - 第二天：项目监控、裁剪、风险管理
 - 第三天：软件度量及量化管理（可能部分内容会在第二天进行，看课程进度）
-

项目大纲 – Statement of Work

- 项目大纲是简要的对项目的描述
 - 项目大纲可能会以不同的名字出现，诸如工作陈述、工作要求或项目建议书
 - 可以来自组织内部，也可以来自外部
 - 谁发起的这个项目？
 - 项目的发起人是否明确了这个项目要做什么？
 - 项目大纲用于业务决策支持
 - 管理层对项目大纲的批准可以看作对项目活动开始的授权（例如需求定义活动）
-

项目大纲可能包含的内容

- 项目要达成的技术和管理目标
 - 项目的客户、用户和环境
 - 项目要解决的问题
 - 预期的收益
 - 限制（成本，日期，资源等）
 - 与其他项目的依赖关系
 - 项目的关键职责
 - 项目的性能目标（质量、生产率等）
-

#讨论

- 为什么需要做规模估计？
 - 估算成本和工作量的输入
 - 度量的需要，可以分析项目的性能
 - 和其他项目建立可比性
 -
 - 如何做好规模估计？
 - 1 有历史数据
 - 2 采用多种方法互相验证
 - 3 任务分解的颗粒度要适当
 - 4 确保没有遗漏
 - 5 有经验的人参与估算
 -
-

阶段描述（一）

- 对生命周期的每个阶段，要按以下要素进行描述：
 - 目的：描述阶段的目的是什么
 - 流程图：用流程图描述活动之间的关系，图中标明每个活动的输入、输出等
 - 角色与职责：描述该阶段中参与的主要角色，并分别描述在该阶段中的角色担负的主要职责
 - 入口准则：进入该阶段必须具备的前提条件。如执行该阶段的角色应具备的能力和资源、约束条件已满足等
 - 输入：该阶段开始的所有输入，如：执行该阶段的前一个阶段的输出，及这个阶段需要的其他输入等
 - 活动：描述该阶段的主要活动。对每项活动，要描述活动的目的、活动提交的工作产品、完成准则等
-

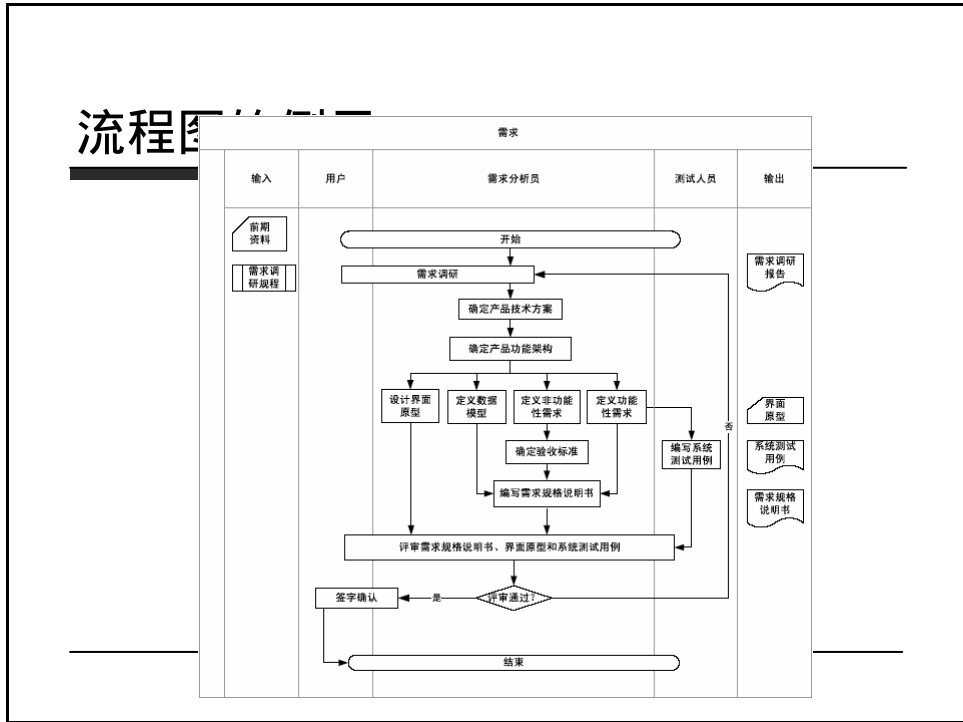
阶段描述（二）

- 输出：说明该阶段的主要输出
- 出口准则：说明该阶段结束的标准，如该阶段完成的标志、完成的约束条件等
- 可应用的标准、规范、模板、指南等：说明该阶段应该遵循的标准和规范等
- 可应用的技术、方法和工具等：说明该阶段要使用到的技术、方法和工具等
- 度量与分析：说明该阶段要进行哪些度量与分析

阶段描述的例子-需求阶段

目标	完成一个可跟踪的需求分析需求的文档
适用标准和规范	XX公司需求开发、需求管理过程定义及相应规程
相关工具	
主要输入	1) 客户需求/需要 2) 项目计划
入口准则	1) 项目计划得到评审和批准 2) 项目策划阶段已经结束 3) 参与需求分析的人员接受过相关技能的培训
参与人员和相关人员	高层经理、项目经理、需求分析师、设计人员、测试人员、PPQA、CM工程师、客户或客户代表、领域专家和技术专家
活动	1. 准备需求采集和分析 2. 采集和分析需求 3) 准备SRS 4) 细化需求跟踪矩阵 5) 计划系统测试 6) 评审SRS、系统测试计划和测试用例、需求跟踪矩阵
主要输出	1) SRS 2) 用户需求说明书 3) 需求跟踪矩阵 4) 系统测试计划和测试用例
出口准则	SRS、系统测试计划和测试用例、需求跟踪矩阵得到评审和批准并置于配置管理之下
度量	需求分析所花的工作量和资金，评审工作量和返工工作量

流程图



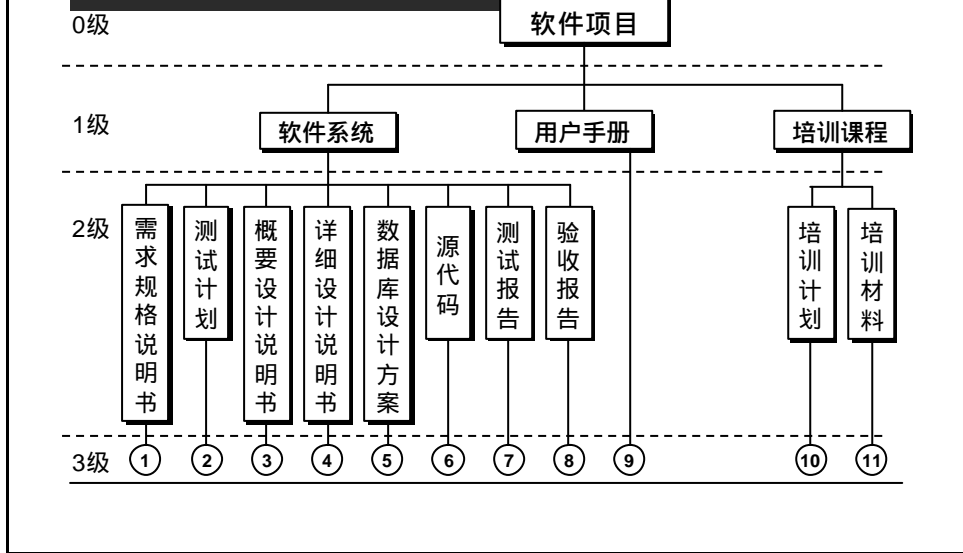
#IBM公司建议WBS分为5层

- L1: 项目
- L2: 可交付物, 如软件, 收集和培训课程
- L3: 构件, 是产生可交付物需要的关键工作项, 如产品系统软件需要的模块和测试
- L4: 工作包, 产品构件需要的主要工作项或相关任务集
- L5: 任务, 通常由单个人完成

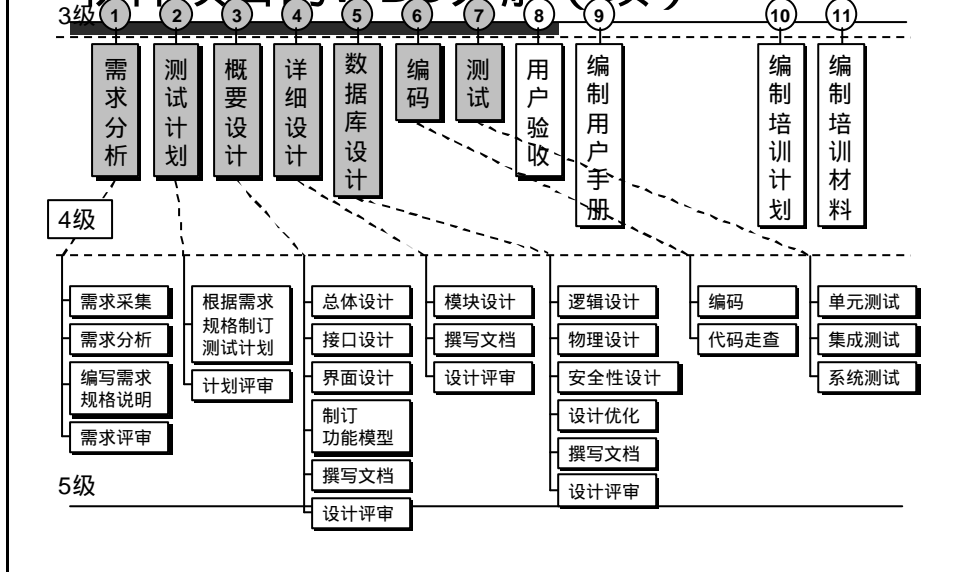
摘自 <软件项目管理>

(英)Bob Hughes, Mike Cotterell 著, 周伯生, 廖彬山, 任爱华翻译

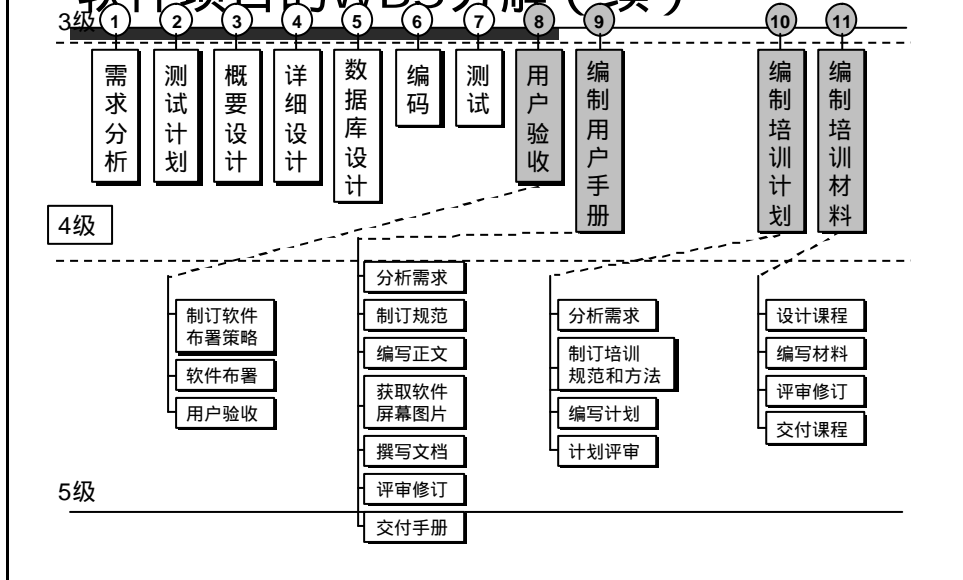
软件项目的WBS分解



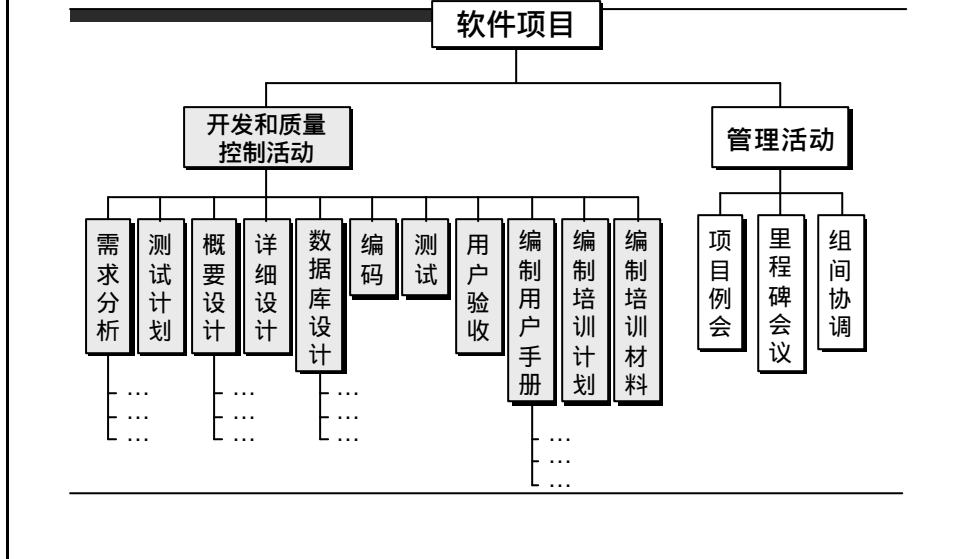
软件项目的WBS分解 (续)



软件项目的WBS分解 (续)



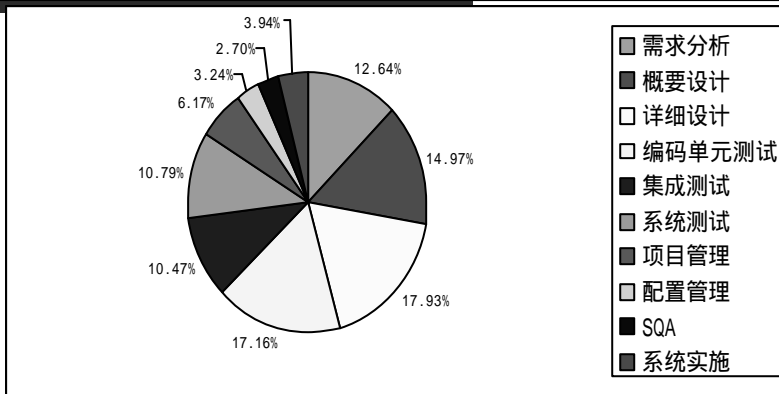
软件项目的WBS分解 (续)



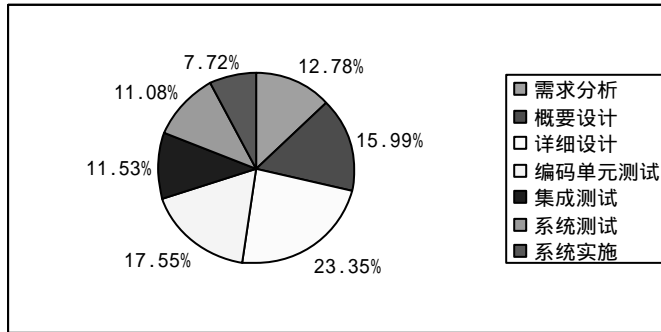
在定义WBS时需考虑的问题

- 既要按活动进行分解（纵向分解），又要按产品进行分解（横向分解）
 - 软件项目过程中包括以下几类活动：
 - 开发活动（如需求分析、设计、编码等）
 - 质量控制活动（如评审和测试）
 - 管理活动（如项目例会、里程碑会议、组间协调等）
- 分解到子节点时，分解要合理，即“粒度适当，大小可比”
- WBS（和项目计划）都要遵循不断细化的原则
 - 迭代式构造
 - 建立逐步细分的层次结构

按任务类型的工作量分布图



按阶段的工作量分布图



阶段工作量和进度周期的分布

阶段	工作量 (%)	进度 (%)
计划和需求	(+8)	(+36)
产品设计	18	36
详细设计	25	18
代码和单元测试	26	18
集成和测试	31	28

COCOMO模型

按活动类型的分布

活动	预算 (%)
需求分析	4
产品设计	12
编程	44
测试计划	6
确认和验证	14
项目职责	7
配置管理和质量保证	7
手册编写	6

COCOMO模型

估算过高和估算过低

- 帕金森定律 (Parkinson's Law)
 - “工作总是用完所有可利用的时间 (Work expands to fill the time available)”
- 布鲁克斯定律 (Brooks' Law)
 - 实现一个项目需要的工作量不是与分配到项目的员工数成比例地增长。当项目组的规模增长时，投入管理、协调和沟通的工作量也在增长
 - 极端情况下，Brooks定律会出现这样的情况：“投入更多的人到一项延迟的工作上，可以导致该项工作更加延迟”

多重回归的示例

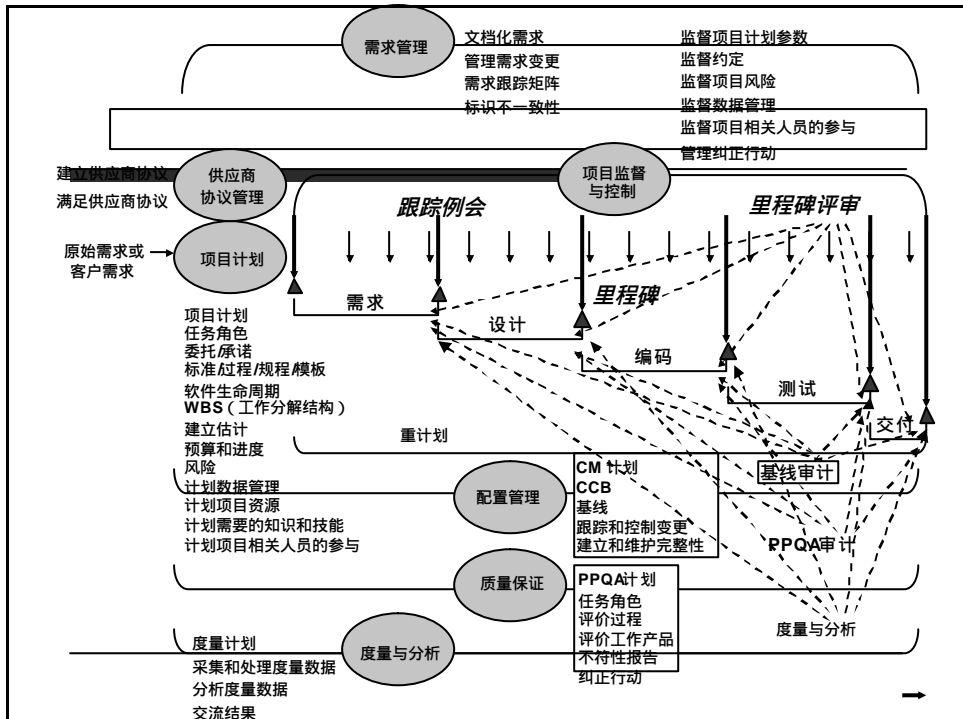
- 假定要编写一个程序：估算大概需要650行的新代码，还计划从以前程序中不加修改重用3000行代码，同时修改该程序中的其他155行代码，以便重用
- 历史数据如后面的表所示：
- 利用这些信息如何计算出开发新代码需要多少时间、重用以前的程序需要多少时间、修改代码需要多少时间？
- 此时，可以用3个变量、4个 的回归公式为：
 - $Hours_t = a_0 + New_k * a_1 + Reuse_k * a_2 + Modified_k * a_3$
 - 其中，下标t表示开发时间的总估算，下标k表示最近的估算值
- 通过解多重线性回归方程得到参数 如下：
 - $a_0 = 6.701$; $a_1 = 0.0784$; $a_2 = 0.0150$; $a_3 = 0.2461$
- 将值代入上面的公式中，得到新项目的估算开发小时数是140.902
- 对参数 可以这样理解：
 - 每个项目有花费时间的常量，即约为6.7小时
 - 开发一行新代码，需要0.0784小时，即每小时12.76行
 - 重用一行代码，需要0.015小时，也就是每小时66.48行
 - 修改一行代码，需要0.246小时，也就是每小时4.06行

多重回归的历史数据

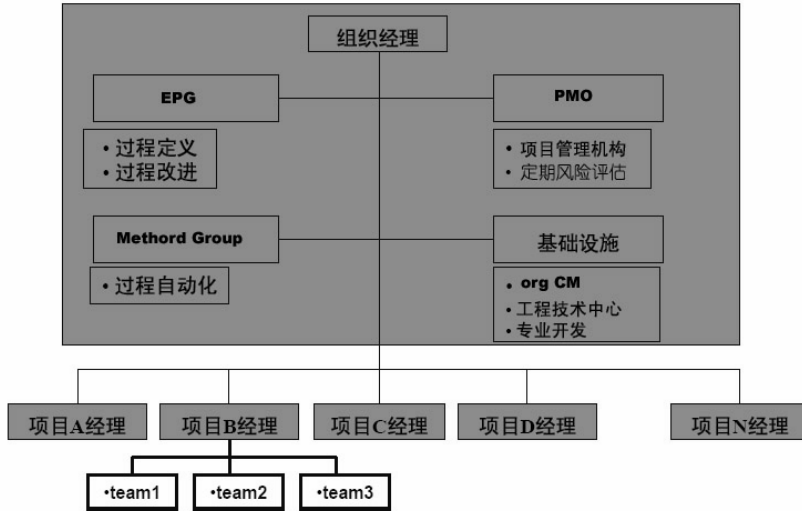
程序编号	新LOC	重用LOC	修改LOC	小时数
1	1142	1060	325	201
2	863	995	98	98
3	1065	3205	23	162
4	554	120	0	54
5	983	2896	120	138
6	256	485	88	61
总计	4863	8761	654	714
均值	810.5	1460.7	109	119
要开发程序的估算值	650	3000	155	140.90

培养承诺的文化

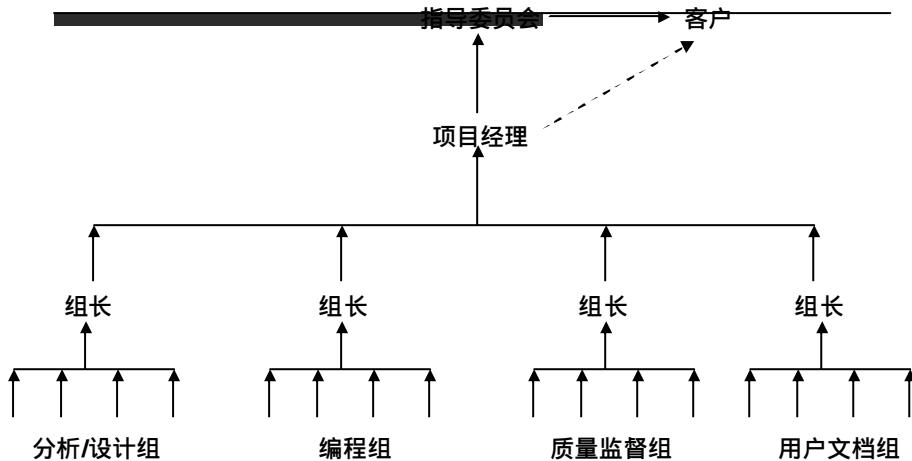
- 承诺必须是自愿的
- 承诺必须是透明的,必须是公开的
- 承诺及其变更必须和相关人员进行了充分协商
- 承诺必须是经过充分准备的,避免过度承诺
- 承诺要随时而变,当无法实现时要提前告知
- 承诺的实现过程要透明
- 为了承诺要尽力而为



组织和主任的例子



项目的报告结构



日报 ???

- 日报的填写是任务
 - 每个任务该填写什么信息（属性，字段）？
哪些信息在制订计划时填写的，哪些信息是由每个执行人在填写日报填写的？
 - 设计一个日报的表格格式
 - 如果项目结束时，想自动得到每个阶段的工时占比，不同类型任务的工时占比，每个任务要有什么信息
-

关于任务与日报（没有工具）

- 每个任务
 - 计划阶段(在进度表中)：编号、任务名称、任务描述、责任人、任务所在阶段、任务的类型、计划的工时、计划的开始时间、计划的结束、依赖的任务、是否返工
 - 实际执行时（在进度表中）：实际的工时、实际的开始时间、实际的完成时间、完成比例、偏差比例
 - 实际执行时（在日报中）：每个任务每天投入的实际工时，是否完成，存在的问题
-

工作日志

软件部员工工作周报											
部 门						姓 名	张某		日期	2007-1-8 到 2007-1-12	
注意事项: 每周末进行本周工作总结。											
	任务名称	项目名称/ 管理事项	阶段	任务类型	任务描述	计划工 时(h)	实际工 时(h)	是否完成	进度状态	备注	
计 划 内 工 作	1										
	2										
	3										
	4										
	5										
计 划 外 工 作	1										
	2										
	3							1			
	合 计						0	0			
总 计						0	0				
本周工作 心得体会 与建议: <hr style="width: 80%; margin-left: 0;"/>											

日志填写与统计 - Microsoft Internet Explorer
_ | 0 | X

文件(F) 编辑(E) 查看(V) 收藏(A) 工具(T) 帮助(H)
地址(D) http://192.168.1.39/DPMPSETUP/DATAWORK/default.aspx

日志填写与统计系统

当前登录的帐号: chenlian 时间: 2005-6-3

[日志填写](#) | [日志修改](#) | [日志查看](#) | [注销](#) | [帮助](#) | [修改密码](#)

已立项 (计划内任务 计划外任务) 未立项

项目 名称: 项目规模: 大型 项目类型: 开发型产品类自主开发MC 项目生命周期模型: 瀑布模型 当前阶段: 概要设计 项目结果: 未结束项目尚未定论

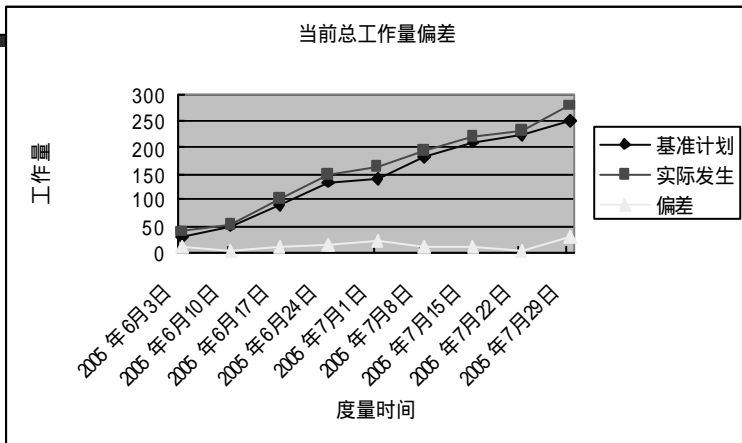
阶 段: 活动类型: 工作时间: (小时)

工作包名称: 完成 未完成

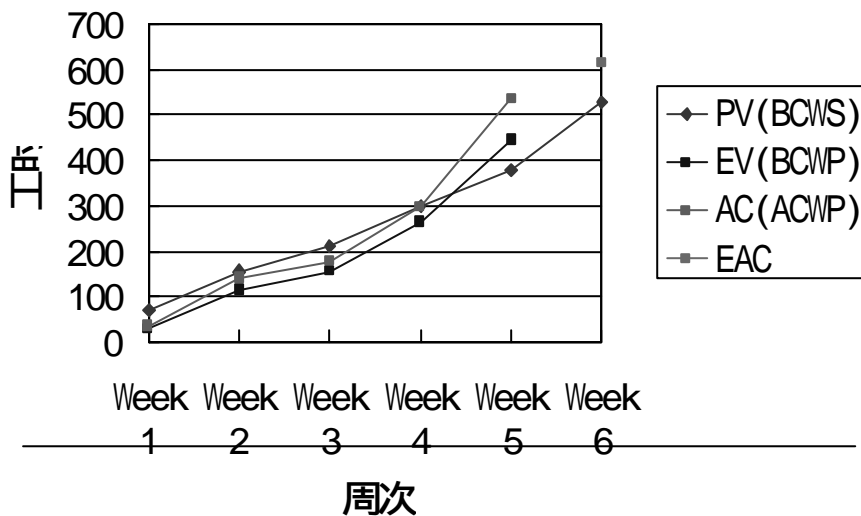
具体描述:

2005-6-3 已填写的日志如下所示:

项目名称	阶段	活动类型	活动描述	工作时间



挣值分析 - 例图 挣值图



挣值(earned value) -1

- 挣值分析近年来得到了广泛应用。它基于对初始花费的预测，赋给每个任务或工作包（在WBS中确定的）一个“值”，这个值是该任务的初始预算成本，称为基线预算（planned value PV）或已计划工作的预算成本（budgeted cost of work scheduled BCWS）。挣值可以用货币价值度量，但对于人力密集型项目，如软件开发项目，通常用“人时”或“人日”来度量
- 一个任务在尚未开始之前用零作为初值，当该任务完成时，则用该任务的赋值作为任务的得分（因而也是项目的得分）。项目在任何点得到的总值称为挣值（earned value, EV）或者已执行工作的预算成本（budgeted cost of work performed BCWP）
- 在任务已开始但尚未完成的情况下，必须采用一致的给挣值赋值的方法，在软件项目中，常见的方法有：
 - 0/1法：只要任务未完成，便赋其值为零，一旦完成任务，便将预算值的100%赋予它
 - 50/50法：任务一开始执行，就赋予任务预算值的50%，一旦任务完成便给予任务100%的赋值
- 每一项任务的实际成本被称为实际成本（actual cost AC），也可被称为已执行工作的实际成本（actual cost of work performed, ACWP）

挣值 -2

- 进度偏差（Schedule variance SV）= EV - PV
 - 表示已完成的工作值与计划的工作值不同的程度
 - 正的CV值，表示项目进度提前；负的SV值，表示项目进度滞后
- 成本偏差（Cost variance CV）= EV - AC
 - 表示已完成工作的预算成本与已完成工作的实际成本之间的偏差
 - 正的CV值，表示低于预算成本；负的CV值，表示超出预算成本
- 进度性能指标
 - $SPI = (BCWP - BCWS) / BCWS$ 或 $SPI = (EV - PV) / PV$
 - 新版计算公式： $SPI = EV / PV$
- 成本性能指标
 - $CPI = (BCWP - ACWP) / ACWP$ 或 $CPI = (EV - AC) / AC$
 - 新版计算公式： $CPI = EV / AC$
- SPI和CPI应该同时接近0。负的结果表明项目落后于进度或超出预算成本。正的结果说明项目比进度提前或低于预算成本

挣值法：讨论与练习

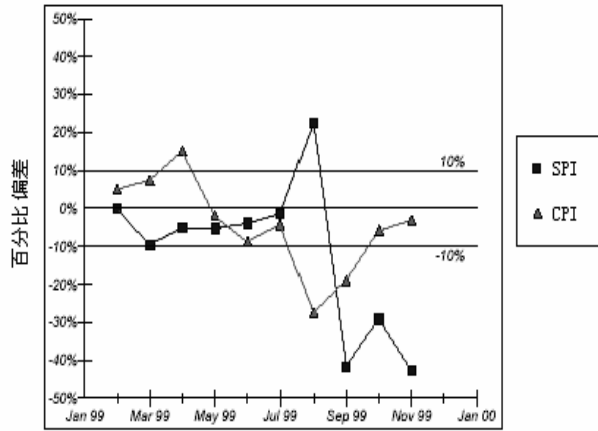
- ~~某土方工程总挖方量中1万立方米。预算单价为45元/立方米。该挖方工程预算总费用为45万元。计划用25天完成，每天400立方米~~
 - 开工后第7天早上刚上班时，进行测量，取得2个数据：已完成挖方2000立方米，支付给承包单位的工程进度款累计已达12万元
 - 计算：
 - BCWS：已安排工作的预算支出（PV）
 - BCWP：执行工作的预算支出（EV）
 - ACWP：执行工作的实际支出（AC）
 - $SPI = (BCWP - BCWS) / BCWS = (EV - PV) / PV$
 - $CPI = (BCWP - ACWP) / ACWP = (EV - AC) / AC$
-

挣值法：答案

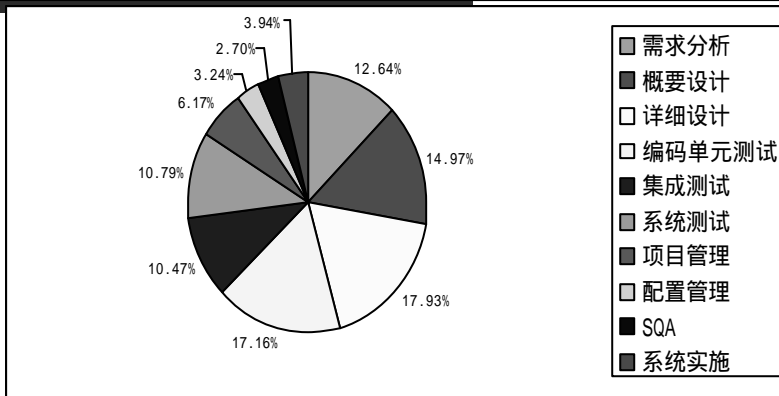
- 已完成工作预算：
 - ~~BCWP = 45元/立方米 * 2000立方米 = 9万元~~
 - 按计划，第6天结束时工作预算：
 - $BCWS = 45\text{元/立方米} * 400\text{立方米/天} * 6\text{天} = 10.8\text{万元}$
 - 实际工作成本：ACWP = 12万元
 - 成本偏差：
 - $BCWP - ACWP = 9\text{万元} - 12\text{万元} = -3\text{万元}$
 - 表明承包单位已超支
 - 进度偏差：
 - $BCWP - BCWS = 9\text{万元} - 10.8\text{万元} = -1.8\text{万元}$
 - 表明进度落后1天： $-1.8\text{万元} / 45\text{天} / 400 = -1$
 - 成本性能指标CPI
 - $CPI = (BCWP - ACWP) / ACWP = -3 / 12 = -0.25$
 - 进度性能指标SPI
 - $SPI = (BCWP - BCWS) / BCWS = -1.8 / 10.8 = -0.167$
-

财务性能：挣值

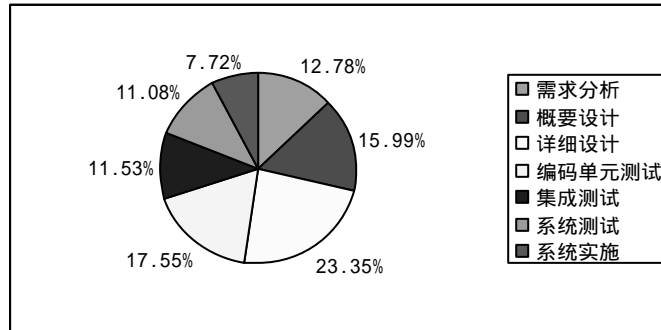
- 左边的图警告项目在早期就已经落后进度并超出成本
- 在8月份的进度差异中，大的峰值应该已经得到了调查。它可能与实际的核算过程中存在的问题有关。三个月后能定的是决策的阈值，表明项目进度落后，应该加以调查。重要的是



按任务类型的工作量分布图



按阶段的工作量分布图



资源模型

□资源模型是指：

- 生命周期各个阶段的工期 (Duration) 以及这些工期与生命周期总工期之比
 - 生命周期各个阶段所花费的工时 (effort) 以及这些工时与生命周期总工时之比
 - 各种不同类型的任务所花费的工时以及这些工时与生命周期总工时之比
 - 生命周期某个阶段 (时间段) 内不同工种所花费的工时以及这些工时与该阶段所花费的工时之比
- 通过填写工作日志，可以建立资源模型
- 坚持“日填、日(抽)查、周报、周/月分析”的制度
- 数据采集时要注意以下各点
- 有用性、真实性、及时性、科学性、简易性、半自动化/自动化
- 按项目规模、类型及所用的生命周期等来建立组织级的资源模型
- 不断积累数据，并根据积累的数据定期对组织级资源模型做更新

任务类型

项目策划(包含量策划): 估计和制订计划等

项目监控: 包含开发人员写周报、项目经理写周报、查工作日志、查进度表、阶段状态报告、问题跟踪、处理邮件、人员协调等

会议: 包含周例会、临时会议、里程碑会议等项目的会议

评审: 包含个人预评审、走查、正式评审会议、审查等(准备时间、会议时间、返工时间、规模、缺陷数等)

配置管理

PPQA (过程和产品质量保证): QA人员专用。QA人员找出的问题, 相关人员处理时填写到相关的任务类别中

培训: 包含自学。(分接受培训、提供培训。只关注项目级)

返工(只有一种, 不细分)

过程改进: EPG人员专用

环境搭建

项目总结: 包含开会、写总结报告、收集项目数据等

需求开发(RD)

系统分析(SA)

系统设计(SD)

功能设计(FD)

详细设计(DD)

集成(SI)

美工

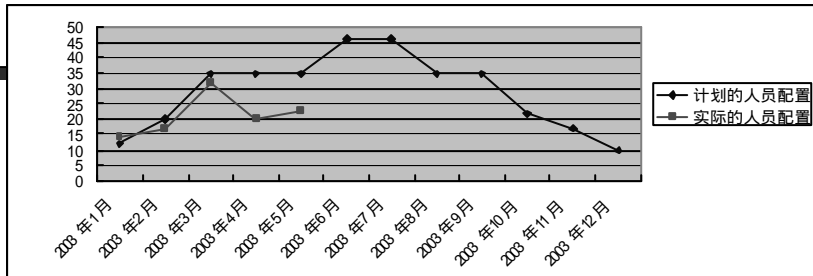
编码(P)

单元测试(T1) (把测试工作T1、T2、T3细分为编写测试用例、编写测试代码、实施测试、缺陷修复、测试其它)

集成测试(T2)

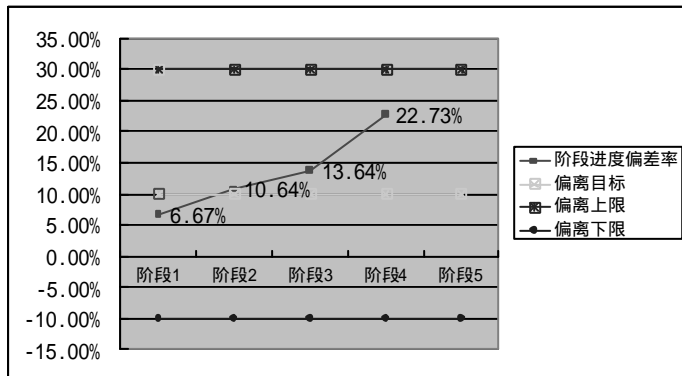
系统测试(T3)

人员配置情况



	1-Jan	1-Feb	1-Mar	1-Apr	1-May	1-Jun	1-Jul	1-Aug	1-Sep	1-Oct	1-Nov	1-Dec
计划的人员配置	12	20	35	35	35	46	46	35	35	22	17	10
实际的人员配置	14	17	32	20	23							
人员配置差异	17%	-15%	-9%	-43%	-34%							

例子：进度监控



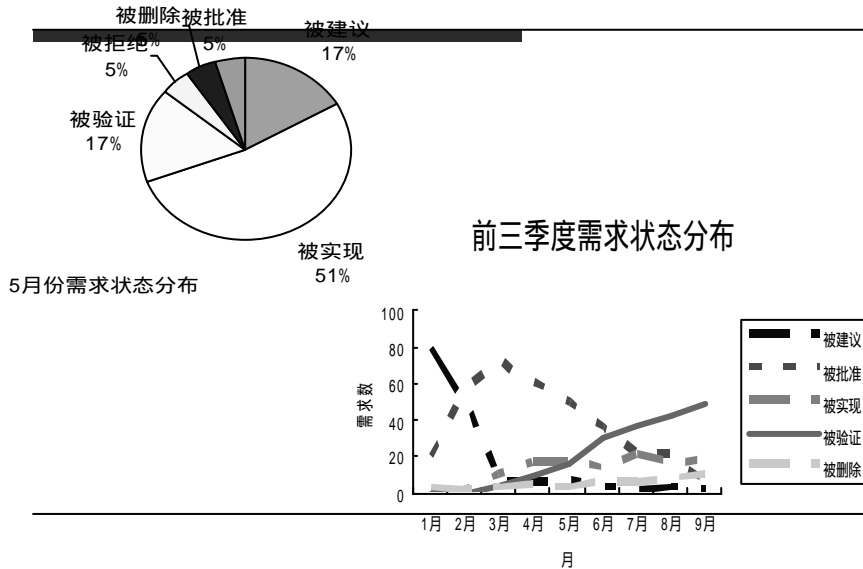
issue :

进度持续延迟，预测点显示下阶段进度趋势接近控制上限—临界状态；

需求状态

- 一个需求经提出、评审、设计、编码实现和测试，它的状态将发生变化
- 可以将需求分被建议、被拒绝、被批准、被实现、被验证、被废除、被交付等状态
 - 被建议：该需求已被有权提出需求的人建议
 - 被拒绝：该需求被建议后，评审未通过
 - 被批准：该需求已被分析，估计了其成本和对项目其他部分的影响，而且通过评审
 - 被实现：已实现需求的设计、编码和单元测试
 - 被验证：使用所选择的方法（如测试）已验证了实现的需求，该需求现在被认为完成
 - 被废除：被批准的需求已从基线中废除，但记录了原因说明和做出废除决定的人员
 - 被交付：需求已通过客户的验收

对需求状态的跟踪

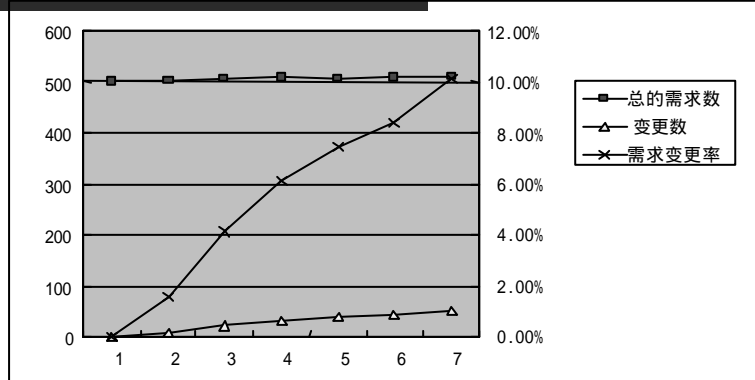


需求稳定性的跟踪

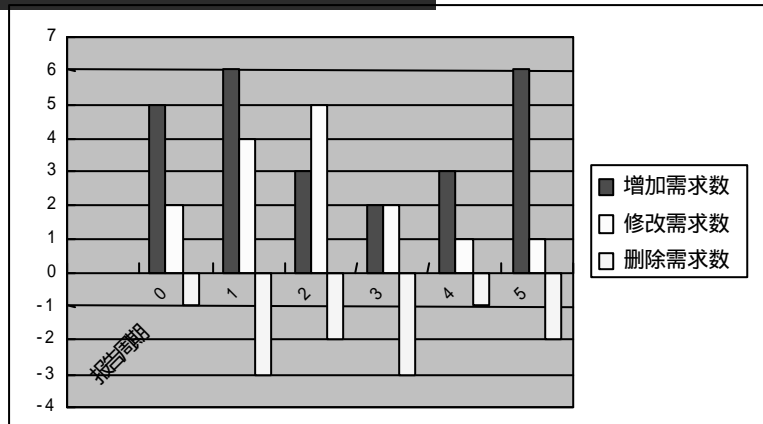
报告周期	增加需求数	修改需求数	删除需求数	总的需求数	变更数	需求变更率
0	0	0	0	500	0	0.00%
1	5	2	-1	504	8	1.60%
2	6	4	-3	507	21	4.20%
3	3	5	-2	508	31	6.10%
4	2	2	-3	507	38	7.60%
5	3	1	-1	509	43	8.60%
6	6	1	-2	513	52	10.40%

- 总的需求数 = (基线需求数 + 增加的需求数 - 删除的需求数)
- 需求变更数 = (增加的需求数 + 删除的需求数 + 修改的需求数)
- 需求变更率 = 需求变更数 / 基线需求数
- 需求稳定性 = 1 - (需求变更数 / 基线需求数)

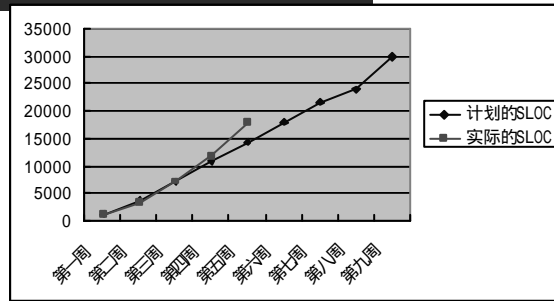
需求变更率



按变更类型的需求稳定性



物理规模和稳定性



	第一周	第二周	第三周	第四周	第五周	第六周	第七周	第八周	第九周
计划的 SLOC	1200	3800	7200	10800	14400	18000	21600	24000	30000
实际的 SLOC	1190	3500	7250	11900	18000				
比率	0.99	0.92	1.01	1.1	1.25				

项目进展报告（如每周）

- 报告每周任务的完成情况：项目组成员把每天的任务完成情况上报给项目经理，对完成的任务进行检查。若完成，会在进度表中说明已经完成。项目经理在写周报时，主要是列出没有完成的任务及偏差情况（内部任务、外部任务）
- 需要的话，报告每个关键任务的偏差天数
- 对工作量的报告，每周的挣值图（来自度量分析报告）。若产生偏差，分析原因
- 对大的项目，报告人员配情况
- 需求变更率、需求稳定性（总的，核心需求（可选））
- 需求状态跟踪
- 对于物理规模的跟踪。若没有计划到每周写多少文档或代码，则不用跟踪；若计划了每周写多少文档或代码，则需要跟踪，出规模跟踪的趋势图
- 对资源获取及情况的跟踪（设备到位情况，计划该到的没有到封面果特别说明）
- 对人员及知识技能获取情况的跟踪（该做的培训没有做，该招聘来的人没有招聘到，该调配到的人没有调配到）
- 数据资料的跟踪（这件事移交给配置管理去做，一般配置管理来报告）
- 对GAN系人及参与情况的跟踪（该周GAN系人参加的事参加了没有）
- 对风险的状态，对缓解措施采取的效果，识别可能发生的新风险
- 总结：该周需要解决的问题及拟采取的措施
- 安排下周的计划

讨论: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
-

阶段总结报告

- 阶段完成情况的总结
 - 阶段进度的跟踪:报告相对偏差、绝对偏差(阶段进度偏差的趋势图)
 - 对工作量的跟踪:按阶段的挣值图
 - 需求变更率与稳定性(按阶段)
 - 需求状态跟踪(按阶段)
 - 对物理规模,报告计划的规模、实际的规模及偏差
 - 跟踪资源到位情况
 - 跟踪知识技能的获取情况
 - 对风险的状态,对缓解措施采取的效果,识别和分析新的风险
 - 对成本进行跟踪(财务)
 - 对GAN系人的跟踪
 - 对阶段的目标(过程性能目标和质量目标)
 - QA的工作情况
 - CM的工作情况
 - 总结:对阶段的问题及解决措施
 - 下一阶段的计划
-

项目总结报告

- 项目的特征信息
 - 项目工作成果的说明
 - 规模：每次计划的估算规模、实际的规模，报告规模偏差率
 - 工作量：每次计划的工作量、实际的工作量，偏差率
 - 各阶段的工作量：计划的，实际的，偏差率
 - 成本：每次计划的成本，实际的，偏差率
 - 进度：计划的工期，实际的工期，工期偏差率
 - 各阶段的工期：计划的，实际的
 - 缺陷：各阶段引入的缺陷数，各阶段排除的缺陷数
 - 生产率
 - 缺陷率
 - 需求变更率与需求稳定性
 - 资源模型的四个分布
 - 对质量目标和过程性能目标是否达成的总结
 - 项目移交的资产
 - 经验教训的总结
 - 最佳实践、样例文档
 - 对项目问题和风险及采取的有效措施进行总结
 - 改进建议
 - 复用资源
-

项目的问题包括哪些？

- 收集要分析的问题
 - 通过评审和执行其它过程来收集问题
 - 要收集的问题包括：
 - 在执行验证和确认活动中发现的问题
 - 项目计划参数与项目计划中的估计的严重偏离
 - 没有满足的承诺（包括内部承诺和外部承诺）
 - 风险状态的严重变化
 - 数据访问、收集、保密性或安全性问题
 - 相关人员标识和参与的问题
 - 分析问题来确定需要的纠正行动
 - 当有遗留的问题或可能阻碍项目满足目的的问题发生时，要求采取纠正行动
-

■ 潜在纠正行动的例子包括：

- 修改工作陈述
- 修改需求
- 修订估计和计划
- 重新磋商承诺
- 增加资源
- 变更过程
- 修订项目风险

P
M
C
与
P
P
的
对
照

PP的特定实践	PMC的特定实践	PMC中子实践
SP1.1 估计项目的范围		
SP1.2 建立项目属性的估计	SP1.1 监督项目计划的参数	监督工作产品和任务的属性
SP1.3 定义项目生命周期		
SP1.4 确定工作量和成本的估计	SP1.1	监督项目成本和所花的工作量
SP2.1 建立预算和进度	SP1.6 执行进展评审 SP1.7 执行里程碑评审	按进度监督进展
SP2.2 标识项目风险	SP1.3 监督风险进行	
SP2.3 计划数据的管理	SP1.4 监督数据管理	
SP2.4 计划项目的资源	SP1.1	监督提供和使用的资源
SP2.5 计划所需的知识和技能	SP1.1	监督项目人员的知识和技能
SP2.6 计划项目相关人员的参与	SP1.5 监督项目相关人员的参与	
SP2.7 建立项目计划		
SP3.1 评审影响项目的计划		
SP3.2 协调工作和资源		
SP3.3 获得计划的承诺	SP1.2 监督承诺	
	SP2.1 分析问题	
	SP2.2 采取纠正行动	
	SP2.3 管理纠正行动	

关于裁剪-1

□ 裁剪的含义：

- 增加
 - 删除
 - 替换方法或格式
 - 修改顺序
 - 多选一
 - 修改权限、级别
 - 等等
-

关于裁剪-2

□ 裁剪的对象：

- 生命周期模型
 - 过程
 - 活动
 - 方法
 - 度量指标
 - 质量目标
 - 控制权限
 - 评审方式
 - 活动频率
 - 参考的度量数据、基线
 - 等等
-

关于裁剪-3

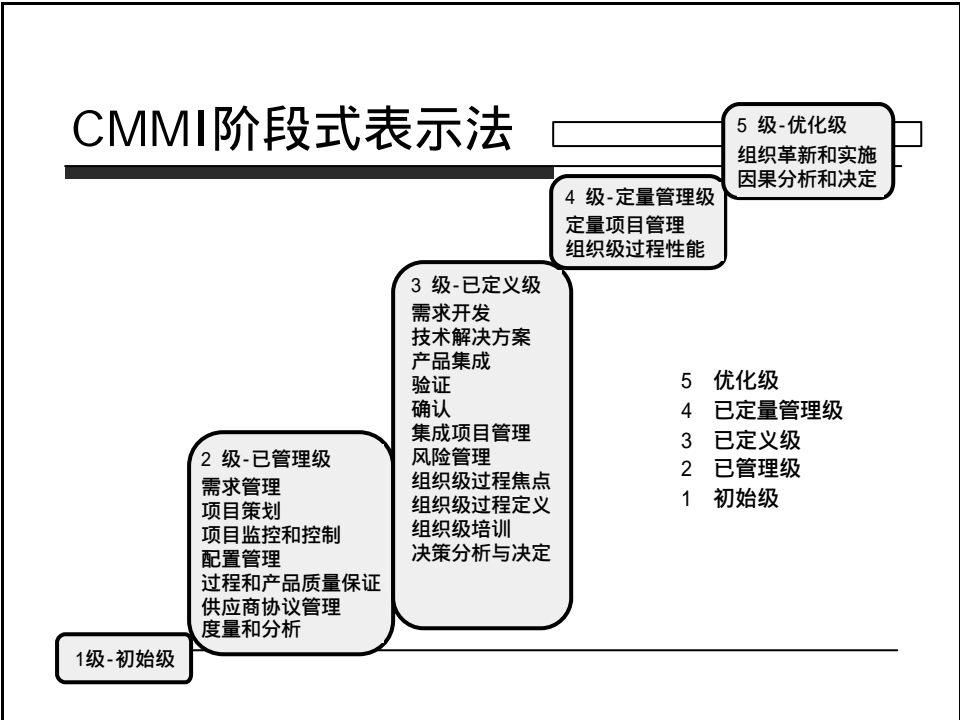
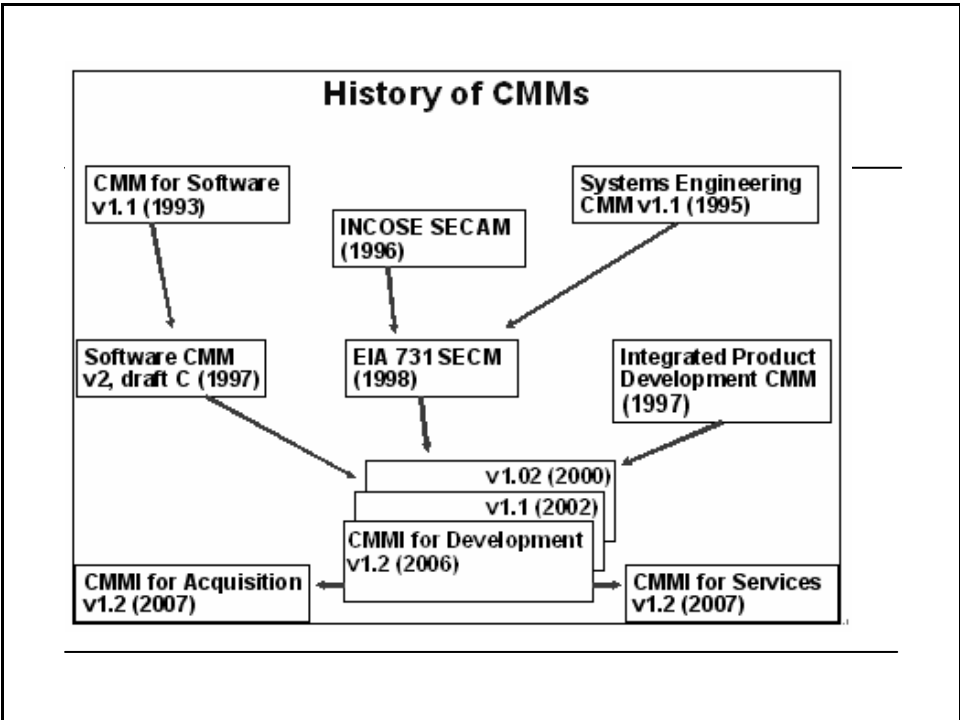
□ 裁剪的步骤

- 确定项目类型
 - 确定项目的过程性能目标和质量目标
 - 选择生命周期，并定义生命周期
 - 活动裁剪（5W1H：做什么，如何做，谁来做，做到什么程度，何时做）
 - 文档裁剪
 - 度量指标裁剪
 - 其他裁剪
 - 形成项目的已定义过程（PDP）
 - 评审项目已定义过程
 - 发布项目组的已定义过程
 - 在里程碑处或重大的变更时确定PDP的变更
-

关于裁剪-4

□ 如何评价组织级裁剪指南的好坏？

- 描述方式：采用简单易用的图、表等形式展示裁剪的指南
 - 抽象层次：对于整个管理的体系应该从策略层、过程层、活动层、模板层定义裁剪指南
 - 完备：上述提到的裁剪对象、裁剪的方法应都有描述
 - 易用：可以针对不同类型的项目或不同类型的活动提供裁剪后的几套模板
 - 明确：裁剪指南的描述没有二义性，确保减少沟通的误差
-



二级（已管理级）的特点

- 方针政策
 - 计划
 - 提供资源
 - 分配职责
 - 培训
 - 管理配置：对工作产品进行适当等级的配置管理
 - 标识和包括有关的相关人员
 - 监督和控制
 - 客观地评价过程
 - 由高层经理进行评审
-

三级（已定义级）的特点

- 已定义过程是已管理的过程
 - 特点是：组织级。已将管理和工程两方面的过程文档化和标准化，并形成了组织级的过程资产，包括：
 - 组织级标准过程集
 - 标准生命周期描述
 - 组织级的剪裁指南和准则
 - 组织级度量库
 - 组织级过程资产库
 - 组织级工作环境标准
 - 所有项目都使用经批准、剪裁的标准过程来开发和管理
 - 三级时要收集数据，也要使用数据
 - 要改变管理模式，从反应式→预测式管理
 - 具有了预测能力，但这种预测能力是定性的
 - 要根据过去收集的几个数据点（周或月），预测今后的趋势
-

关于度量指标

度量分类	度量指标
进度与进展	包括“里程碑进度偏差”、“关键任务或交付日期是否延期”等
资源与费用	包括“工作量偏差”、“预算和进度目标是否满足”等
产品规模与稳定性	物理规模与稳定性：通过“代码规模”、“接口数”、“组件数”等来度量 功能规模与稳定性：通过“需求数”、“功能变更数”、“功能点数”等来度量
产品质量	功能正确性：通过缺陷、缺陷分布等来度量 可维护性：通过复杂度来度量 效率：通过系统资源的利用率、吞吐量和响应时间等来度量 可用性：通过操作员所犯的错误数来度量 可靠性：通过平均故障间隔时间（MTBF）来度量
过程性能	包括“生产率”、“返工工作量”等度量指标
技术有效性	技术适用性：考查采用的技术满足需求的程度 技术易变性：考查采用的技术是否会造成太多的基线变更
客户满意度	包括“满意度评定”、“客户支持的请求数”、“客户支持的响应时间与解决时间”等

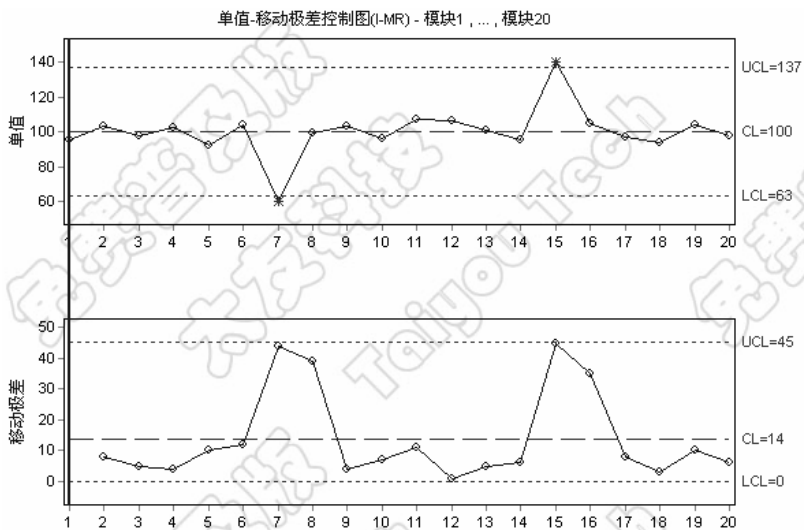
度量指标要解决什么问题？

度量分类	度量指标	要解决的问题
进度和进展	完成的里程碑 关键路径性能 工作单完成进展	项目符合预定的里程碑吗？ 关键任务或交付日期延迟了吗？ 特定的活动和产品进展如何？
资源和费用	人员工作量 财务性能 环境和支持资源	所花工作量是按计划的吗？ 项目是否满足预算和进度目标？ 是否有足够的具备所需技能的员工？需要的设施、设备和材料是否可获得？
产品规模和稳定性	物理规模和稳定性 功能规模和稳定性	产品的规模、内容、物理特性或接口变更有多少？ 需求和相关的功能变更有多少？
产品质量	功能正确性 可维护性 效率 可移植性 可用性 可靠性	产品质量是否达到了交付给用户的水平？已标识的问题解决了吗？ 系统要求多少维护？维护的难度如何？ 目标系统能有效地使用系统资源吗？ 功能在另一平台上重新驻宿达到了什么程度？ 用户界面是足够的且便于操作吗？操作员的错误是在可接受的范围内吗？ 给用户的服务常常被中断吗？故障率是在可接受的范围内吗？
过程性能	过程符合性 过程效率 过程有效性	项目实现已定义过程的一致性如何？ 过程效率是否达到了满足当前委托和计划的目标？（生产率等） 因返工需要花多少额外的工作量？（返工工作量等）
技术有效性	技术适用性 技术易变性	技术满足所有的已分配的需求吗或需要额外的技术吗？ 新的技术是否因太多的变更而造成风险？
客户满意度	客户反馈 客户支持	我们的客户多大程度上理解项目的性能？项目满足用户的期望吗？ 客户支持请求多快能得到处理？

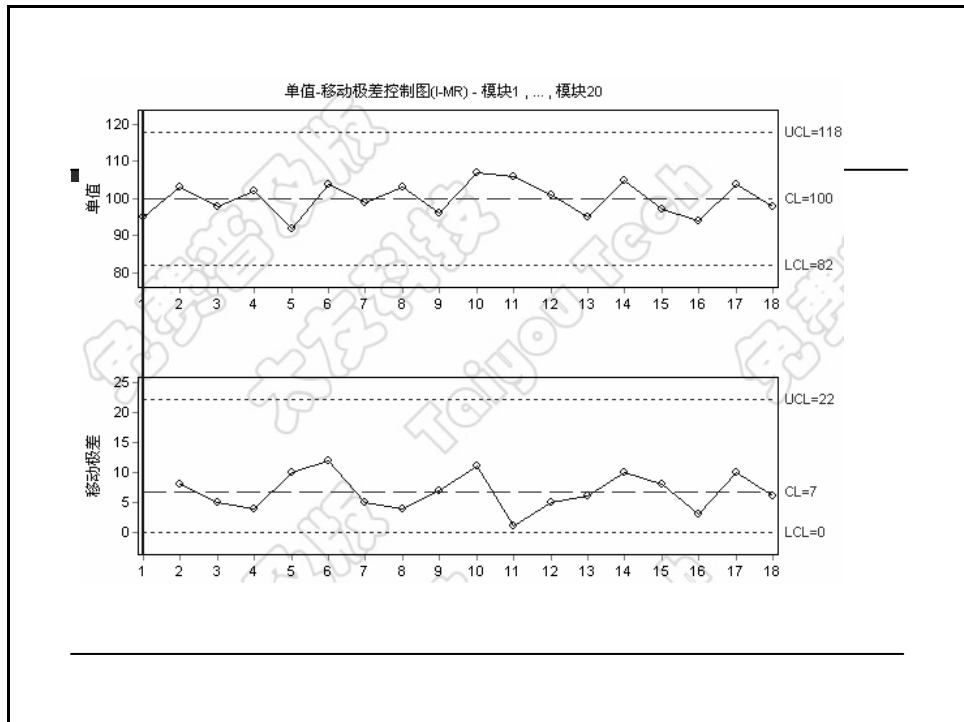
实例：编码阶段的生产率

□ 在编码阶段已完成的20个模块，每个模块的生产率（LOC/人天）如下：

模块1	模块2	模块3	模块4	模块5	模块6	模块7	模块8	模块9	模块10	模块11	模块12	模块13	模块14	模块15	模块16	模块17	模块18	模块19	模块20
95	103	98	102	92	104	60	99	103	96	107	106	101	95	140	105	97	94	104	98



从图中可以看出模块7和模块15的生产率超出了控制限，说明存在特殊原因引起的过程不稳定



对项目级的度量与分析涉及四类人

- 项目组成员: 把做的每件事记下来
- 度量人员: 到时间点(每周/每个月/每个阶段/每半个月等)采集和统计,跟目标比,用指示器去初步分析
- 项目经理: 深度分析,必要的话,采取措施
- 高层经理: 评审和审批

对组织级度量与分析涉及四类人

- 实施人员: 填写基础数据
- 组织级度量人员: 对项目或职能组的数据做采集和统计(如果项目的度量人员已经把项目的度量数据及有关信息已经在项目总结报告中做了统计,则组织级度量人员只需把这些收集到组织级的度量库中). 定期(每季度/半年)根据同类型的数据,形成组织级的指标数据(如组织级平均生产率,上限,下限;平均缺陷率,上限,下限),出一个组织级的度量与分析报告
- EPG(工程过程组)
- 公司高层

度量活动中的角色和职责

高层经理



- ? 确定高层的目标
- ? 使用度量结果做组织级决策

项目经理



- 确定和管理项目的有关事项
- 使用度量结果做出项目决策

度量人员



- 裁剪度量,用来满足需要
- 收集和分析度量数据,并报告结果

开发团队



- 提供度量所需要的数据
- 在软件开发实践中利用度量结果

缺陷及其属性

- 实体：缺陷
- 缺陷处理流程：提交、分析、解决、验证等
- 属性（组1）：
 - 标题（概要描述缺陷）、详细描述（缺陷的重现步骤/截图）、缺陷的来源（缺陷所在的模块）、缺陷的优先级、严重程度、指派PAI给对应的负责人、修复建议（可选项）、发现缺陷的测试用例、缺陷的状态、对应状态的时间、发现缺陷的阶段
 - 产生缺陷的阶段、缺陷修复的办法（解决方案）
- 补充（组2）：
 - 发现阶段：发现人、发现的阶段（如单元测试/集成测试等）、版本信息、缺陷的编号
 - 分析阶段：缺陷的类型、严重程度、影响程度（???）、YU计的修改工作量、需要修改的配置项及其内容、分析的工期、分析的工时、缺陷的表现原因、根本原因、原因分类（???）、缺陷的产生阶段、分析人员、缺陷的解决方案、缺陷处理的优先级
 - 修复阶段：修复的工期、修复的工时、修复人员、修复的规模
 - 验证阶段：验证的工期、验证的工时、验证的人员、回归测试的案例、改缺陷引入的缺陷数量和编号
 - QA投入的工时
- 度量指标：
 - 缺陷的模块分布、严重程度分布、指定到某个人的缺陷的解决状况、缺陷处理的解决周期（从缺陷的提出到关闭的整个周期）、测试用例的有效性、检查表

缺陷及其属性？！

- 实体：缺陷
- 缺陷的属性：
 - 测试问题单（测试问题）/评审报告（评审问题）：
 - 名称、描述、类别*（测试与评审是不一样的）、影响程度（按等级划分）、处理优先级、预计解决工时、实际解决工时、发现人、解决人、解决措施、验证人、状态、引入阶段、排除阶段、缺陷所属的对象（如模块等）、发现方式（评审/单元测试/代码走查/组装测试/系统测试/客户服务/使用等）、对应的检查项（评审）
- 测量：
 - 每周/阶段/里程碑/结项：各类缺陷累计发生/解决/遗留
 - 各类缺陷在引入阶段/排除阶段的分布（对某个项目）
 - 每类缺陷的缺陷率分布

BUG的属性：例子

- ID、项目组
- 现象的具体描述、现象分类
- 出现频次（测了几次出现几次）
- 报告日期、报告人
- 当前的状态（BUG的状态）
- 分配给谁（担任者）
- 优先级
- BUG的完成度（修正/不修正/暂停等）
- 摘要（描述）、附加的信息
- 再现的步骤（在现场描述下面）
- 归属方（内部/外部/编译/测试组等）
- 原因的详细描述（对应问题的原因）、原因分类
- 原因发生的工程（在哪个工程引入的）
- 发现的方法（如单体测度等）
- 发现者、对应者、对应日期、确认者、确认日期
- 是否为BUG？
- BUG的发现版本
- BUG发生的模块
- 最终的解决方法
- 相关的附件（如LOG信息）
- 最后一次的更新日期

缺陷的度量

- 注入阶段分布
- 排除阶段分布
- 类型分布
- 模块/构件分布
- 严重程度分布
- 发现方式（同行评审/测试等）分布
- 缺陷的resided时间 = 发现时间 - 注入时间
- 缺陷的滞留时间 = 排除时间 - 发现时间
- 残留缺陷 = 2 * 最后回归测试中的缺陷
- 注意：要根据二八定理，找出系统中的薄弱环节

缺陷的Pareto分析

- 缺陷的Pareto分析图按成因将缺陷数分级，左端是最普遍发生的缺陷，数据来自于HP公司的4个程序。由图可知，最多的3类缺陷之和占总数的1/3以上，因此确定了过程改进重点，从而完成了重大的质量改进

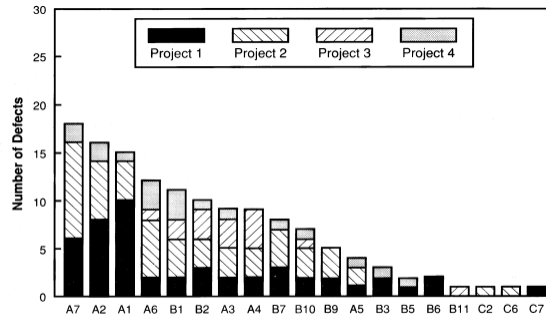


FIGURE 16.3
Pareto Analysis of Software Defects [9]

代码审查九句真言

- 1、看见了if，就想else
- 2、看见malloc，就去找free
- 3、函数调用要小心，需要看看返回值
- 4、看到for循环，就找边界值
- 5、看见return要注意，要去前面找资源
- 6、看见数组把神提，问题往往在下标
- 7、不要小看字符串，长度是个大问题
- 8、得到函数不要急，看看变量初始化，各种路径要小心
- 9、赋值函数最危险，变量没有初始化
- 九句真言不孤立，相互结合显神威

组织级同类型项目的生产率

- 信息需要：
 - (1) 通过已有项目的生产率来了解组织级生产率水平；
 - (2) 利用组织级生产率来预测项目的生产率
 - 指示器：
 - 分析模型：根据类型所有项目的生产率，计算平均生产率和标准差，画出控制图，把每个项目的生产率标在控制图上
 - 控制图
 - 决策准则：参见对控制图的四种测试
 - 派生测量：项目的生产率=项目的规模/项目的工时
 - 基本测量：项目的工时=(每个任务的实际工时累加)，项目的规模=(每个模块的功能数/代码行数/功能点数累加)
 - 实体：
 - 每个任务，属性：每个任务的实际工时
 - 每个模块，属性：每个模块的功能数/代码行数/功能点数
-

挣值图

- 信息需要：了解项目的成本和进度偏差，来控制项目在资源、进度、质量等方面的风险
 - 指示器：挣值图
 - 分析模型：把每周的PV，EV，AC连起来形成三条趋势性
 - 决策准则：当SPI和CPI超出+/-12%，需要分析原因。必要的话，采取措施
 - 派生测量： $SPI = (EV - PV) / PV$ ； $CPI = (EV - AC) / AC$
 - 基本测量：每周累计迄今为止的PV，EV，AC
 - 实体：任务
 - 属性：计划工时(人天)，实际工时，是否完成
-

严重程度：A, B, C, D

刻度类型：标称刻度

刻度：A, B, C, D

计划工时，实际工时

刻度：0-无穷大

刻度类型：区间或比例

测量单位：人月

考虑规程的时候4W + 1H

填写人(基础数据),采集人(基本测量/PAI测量/画出指示器/做初步分析),分析人(针对指示器/深度分析),汇报给谁(针对分析结果,对决策做审批)

填写时间(基础数据,数据填写在哪个表格或文档中),采集时间(每周/双周/每月/阶段点,基本测量数据/PAI生测量数据,在哪个报告中,项目进展报告,阶段报告,总结报告,测试结果的分析报告),分析时间(每周/双周/每月/阶段点,针对指示器,产生分析结果放在什么地方),汇报时间

How: 测量方法\测量函数/分析模型

进度与进展

可测量概念 (度量指标)	要解决的问题	预期的测量
里程碑完成情况	项目是否符合预定的里程碑？	里程碑日期
关键路径性能	关键任务或交付日期是否延迟？	缓冲时间
工作单元进展	特定的活动和产品进展如何？	已跟踪的需求 已测试的需求 已打开的问题报告 已关闭的问题报告 已完成的评审 已打开的变更请求 已解决的变更请求 已设计的单元 已编码的单元 已集成的单元 已尝试的测试用例 已通过的测试用例 已打开的行动项
增量式能力	要交付的能力像在增量式构造和发布中预定的那样吗？	已集成的组件 已集成的功能

资源和费用

可测量概念 (度量指标)	要解决的问题	预期的测量
人员工作量	所花工作量是按计划的吗？ 是否有足够的具备所需技能的员工？	员工水平 开发工作量 经验水平 人员流动
财务性能	项目是否满足预算和进度目标？	BCWS (计划工作的累计预算成本) BCWP (计划工作的累计实际成本) ACWP (已完成工作的累计预算成本) 预算 费用
环境和支持资源	需要的设施、设备和材料是否可获得？	需要的数量 可用的数量 可用的时间 已用的时间

产品规模和稳定性

可测量概念 (度量指标)	要解决的问题	预期的测量
物理规模和稳定性	产品的规模、内容、物理特性或接口变更有多少？	数据库大小 组件数 接口数 代码行数
功能规模和稳定性	需求和相关的功能变更有多少？	需求数 功能变更数 功能点数

产品质量

可测量概念 (度量指标)	要解决的问题	预期的测量
功能正确性	产品质量是否达到了交付给用户的水平？已标识的问题是否得到了解决？	缺陷 缺陷的延续时间
可维护性	系统要求多少维护工作？维护的难度如何？	恢复的时间 圈复杂度
效率	目标系统能有效地使用系统资源吗？	利用率 吞吐量 响应时间
可移植性	功能在另一平台上重新驻宿达到了什么程度？	标准的符合性
可用性	用户界面是否足够且便于操作？操作员的错误是在可接受的范围内吗？	操作员错误
可靠性	给用户的服务常常被中断吗？故障率是在可接受的范围内吗？	平均故障间隔时间 (MTBF)

过程性能

可测量概念 (度量指标)	要解决的问题	预期的测量
过程符合性	项目实现已定义的过程的一致性如何？	基准成熟度评定 过程审计发现
过程效率	过程效率是否达到了满足当前委托和计划的目标？	生产率 周期时间 (Cycle Time)
过程有效性	因返工需要花多少额外的工作量？	已包括的缺陷 逃逸的缺陷 (Defects Escaping) 返工工作量 返工组件

技术有效性

可测量概念 (度量指标)	要解决的问题	预期的测量
技术适合性	技术满足所有的已分配的需求吗或需要额外的技术吗？	需求覆盖
技术易变性	新的技术是否因太多的变更而造成风险？	基线变更

客户满意度

可测量概念 (度量指标)	要解决的问题	预期的测量
客户反馈	我们的客户多大程度上理解项目的性能？项目满足用户的期望吗？	满意度评定、奖励费用
客户支持	客户支持请求多快能得到处理？	支持的请求数、支持时间

组织级同类项目的生产率

- 填写：
 - 每项任务的工时是项目组成员，工作日志
 - 每个模块的源代码文件由配置管理人员纳入基线库
- 采集
 - 采集人，项目的度量人员
 - 采集时机，产品第一上线发布时（总的生产率）（如果是迭代增量式开发时如何处理？）
 - 统计总工时，用工具统计总代码
 - 项目的生产率=规模/总工时
 - 每个项目的生产率数据要纳入组织级度量库（规模、工时、生产率）
 - 不同语言时，应该考虑编码阶段，出不同语言的生产率，总生产率应该考虑语言之间的相互折算（考虑用功能点折算方法作为中间过渡）
 - 验证人，组织级的度量分析师
- 分析
 - 分析人，组织级的度量分析师
 - 分析时间，每半年
 - 分析结果体现在组织级的定期的度量分析报告中
- 汇报：汇报给EPG负责人和高层领导（过程改进委员会）

度量数据采集与存储规程

来源	(1)每个任务的计划工时来自项目进度计划 (2)每个任务的实际工时来自工作日志
填写人	(1)项目负责人 (2)项目组成员
填写时机	(1)每周 (2)每天
填写说明	(1)项目负责人每周制订下周的计划时估计每项任务的计划工时 (2)项目组成员每天填写每项任务投入的工时，并说明每项任务的完成情况
采集人	度量工程师
采集时机	每周
采集说明	度量工程师统计每周迄今为止项目计划投入的工作量和实际所花的工作量，计算工作量偏差率，并填写在“工作量变化趋势”表中
验证人	项目负责人/PPQA工程师
验证说明	对照数据验证检查单检查填写人所填数据和采集人所采集数据的完整性、正确性和一致性

度量分析与汇报规程

分析人	项目负责人
分析说明	度量工程师根据采集到的度量数据，绘制出指示器，并汇报给项目负责人。项目负责人根据指示器对项目的工作量投入情况进行评价，以此评估项目的进度、成本和预算风险。必要的话，采取措施
分析时机	每周
汇报时机	每周
汇报途径	项目负责人把分析结果和拟采取的措施汇报给高层经理，由高层经理进行评审并审批

度量数据采集与存储规程-挣值

来源	(1)每个任务的计划工时来自项目进度计划 (2)每个任务的实际工时来自日报
填写人	(1)项目负责人 (2)项目组成员
填写时机	(1)每周 (2)每天
填写说明	(1)项目负责人每周制订下周的计划时估计每项任务的计划工时 (2)项目组成员每天填写每项任务投入的工时，并说明每项任务的完成情况
采集人	度量工程师
采集时机	每周
采集说明	度量工程师统计每周迄今为止的PV，EV，AC，计算SPI和CPI，并填写在“挣值图”中
验证人	项目负责人/PPQA工程师
验证说明	项目负责人检查每个成员的日报，并检查任务完成情况；PPQA检查进度表中的填写情况，检查日报填写的符合性，抽查度量工程师所做统计工作的正确性、完整性

度量分析与汇报规程-挣值

分析人	项目负责人
分析说明	度量工程师根据挣值图做初步分析，主要是判断是否超出了决策准则规定的进度和成本偏差率，并汇报给项目负责人。项目负责人根据挣值图对项目的成本和进度进行评价，以此评估项目的进度、成本、资源和预算等方面的风险。必要的话，采取措施。分析结果及拟采取的措施，体现在项目周报中
分析时机	每周
汇报时机	每周
汇报途径	项目负责人把分析结果和拟采取的措施通过周报汇报给高层经理，由高层经理进行评审并审批

度量数据采集与存储规程-组织级生产率

来源	(1)每个任务的实际工时来自成员的日报 (2)整个系统的代码行来自于项目代码库中的源程序文件
填写人	(1)项目组成员 (2)项目的度量人员
填写时机	(1)每天 (2)项目结束时
填写说明	(1)项目组成员每天填写每项任务投入的工时,并说明每项任务的完成情况 (2)项目结束时,由项目的度量人员用工具统计每个源程序文件,汇总得到整个项目的规模,填写在项目总结报告中
采集人	度量的度量人员/组织级的度量人员
采集时机	项目结项时
采集说明	项目度量人员统计整个项目的每个任务的实际工时,得到项目的总工时,填写在项目总结报告中;组织级度量人员从项目的总结报告中把规模和工时数据收集到组织级的度量库中,计算项目的生产率
验证人	项目负责人/PPQA工程师
验证说明	项目负责人检查每个成员的日报,并检查任务完成情况;PPQA检查进度表中的填写情况,检查日报填写的符合性,抽查度量工程师所做统计工作的正确性、完整性。对项目度量人员的检查.....

度量分析与汇报规程

分析人	EPG
分析说明	组织级度量人员把迄今为止已完成的同类型项目的生产率,得到平均生产率和标准差,画出控制图,再各项目的生产率标在控制图上,然初步分析后,提交EPG,由EPG做进一步的分析,根据分析结果提出组织下一步的改进建议
分析时机	每季度
汇报时机	每季度
汇报途径	

关于度量体系的建立 -1

- 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
 - 根据度量的信息需要，依照“度量信息模型”的办法，按指示器、派生测量、基本测量、属性进行分解
 - 分解后确定属性的最小集，即利用这些属性就可得到度量的信息需要
 - 分析这些属性属于什么实体，通过填写哪些基本表格能得到？
 - 大致有这些基本表格：工作日志表、缺陷表、变更表等。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
 - 大致有这些报告：项目周报、里程碑报告、项目总结报告、组织级度量库等
-

关于度量体系的建立 -2

- 项目级的度量数据
 - 里程碑完成情况
 - 工作量情况
 - 资源模型（四个分布）
 - 预算和费用
 - 规模情况（代码行数）
 - 需求变更率（需求稳定性）
 - 需求变更的工作量情况
 - 变更引起的增加/减少的工作量
 - 缺陷率（各阶段的、交付时的）
 - 发现的缺陷数/实际总规模
 - 缺陷的分布
 - 缺陷的发现阶段分布、缺陷的模块分布、缺陷的类型分布、缺陷的严重程度分布、缺陷的发现方式分布
 - 缺陷的滞留时间=排除时间-发现时间。计算出各个缺陷在系统中的滞留时间，并按逆序排序
 - PPQA发现的不符合问题
 - 分类统计个数
- 生产率（整个项目的，分阶段的，编写测试用例的）
- 等等

关于度量体系的建立 - 3

- 组织级的度量数据（存放在组织级度量库中）
 - 生产率：多个项目的平均生产率，及上下限（平均值 \pm 3个标准差）
 - 各种语言的纯编码生产率
 - 进度偏差：多个项目的平均值，及上下限
 - 工作量偏差：多个项目的平均值，及上下限
 - 资源模型：多个项目的资源模型折算为组织级的资源模型
 - 规模偏差：多个项目的平均值，及上下限。如果要求项目在开发过程中进行多次估算，则每次估算都有偏差
 - 需求变更率：多个项目的平均值，及上下限
 - 需求变更的工作量占整个项目工作量比例：多个项目的平均值，及上下限
 - 缺陷率（各阶段的、交付时的）
 - 缺陷的清除率（各阶段的、交付时的）
 - 各测试阶段的测试用例密度
 - 文档评审效率
 - 代码评审效率
 - 等等
-

项目的特征信息

- 项目名称、版本
 - 结构：C/S、B/S
 - 硬件平台、开发语言、OS、DBMS、开发工具
 - 团队规模
 - 项目周期
 - 规模
 - 工作量
 - 技术复杂性
 - 人员经验水平
 - 全新开发、承前开发
 - 生命周期模型
 - 应用领域
 - 成功或不成功？（如何判断？）
-



Q&A

