

DINGLI COMMUNICATIONS CORP.,LTD. 鼎利通信 鼎力支持 追求卓越 精益求精



研发项目过程管理

(任务计划管理)

—黄锦阶.....

时间：2009年9月



Win-win Agreement



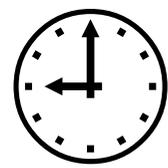
全心参与 由零开始



快乐成长 欢呼收割



请将手机 关机或静音



请您准时 参加培训

课程目标

这课程是针对鼎利项目经理们先打好项目管理的基础，特别是软件开发项目的计划控制内容。领会研发过程管理、项目管理、CMMI、IPD和敏捷开发的思想，再结合鼎利实现情况进行吸收和梳理，把复杂项目管理流程简单化，把简单的流程执行起来，在理解中实施，在实施中理解。



DINGLI COMMUNICATIONS CORP.,LTD.



目录 *Content*

- 一、**研发项目与项目管理的基本概念**
- 二、研发管理、IPD、CMMI、敏捷的定义及关系
- 三、软件开发模型与工作方法
- 四、研发项目管理及组织
- 五、研发项目计划制定
- 六、研发项目计划控制
- 七、在KD研发中心的某一天
- 八、研发项目精细化管理
- 九、KD研发中心项目管理体系

项目的五大特性

目的性

目的性是指任何一个项目都是为实现特定的组织目标服务的。

独特性

独特性是指项目所生成的产品或服务与其他产品或服务都有一定的独特之处

一次性

一次性（也被称为“时限性”）是指每一个项目都有自己明确的时间起点和终点，都是有始有终的，而不是不断重复、周而复始的。

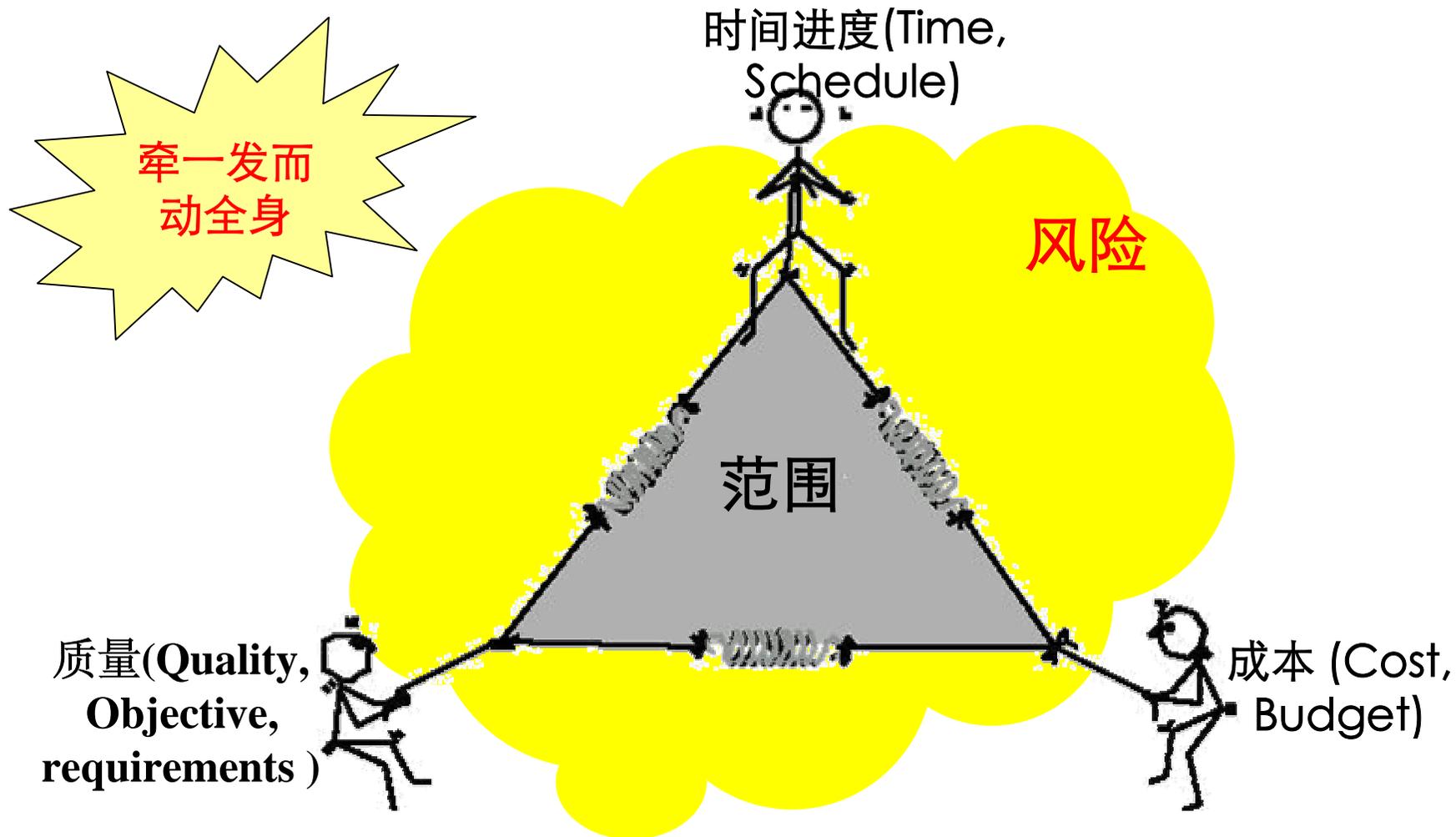
制约性

制约性是指每个项目都在一定程度上受客观条件的制约。最主要的制约是资源的制约。

其它特性

这包括：项目的不确定性、项目的风险性、项目过程的渐进性、项目成果的不可挽回性、项目组织的临时性和开放性等等

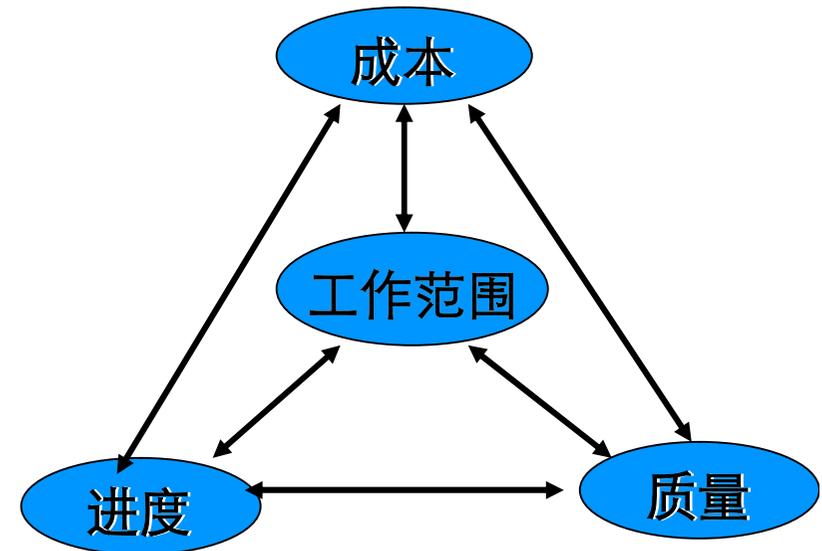
项目的三重目标与三个约束条件



项目成功的含义

项目成功的目标

- S – Scope 在预期的范围内
- T – Time 在预期的时间内
- Q – Quality 达到预期的质量
- C – Cost 在预期的成本内



他们之间的关系是：

$C=f(Q, T, S)$ 改变T、Q、S中的任何一个因素都会引起C的变化

满足或超过项目干系人的期望

客户满意！（Vs 客户接收）

研发项目管理全过程包括以下工作

1. 制定技术目标
2. 组建项目组
3. 制订项目计划
4. 处理范围变化
5. 控制实际进展
6. 整理、完善技术档案
7. 形成知识网络

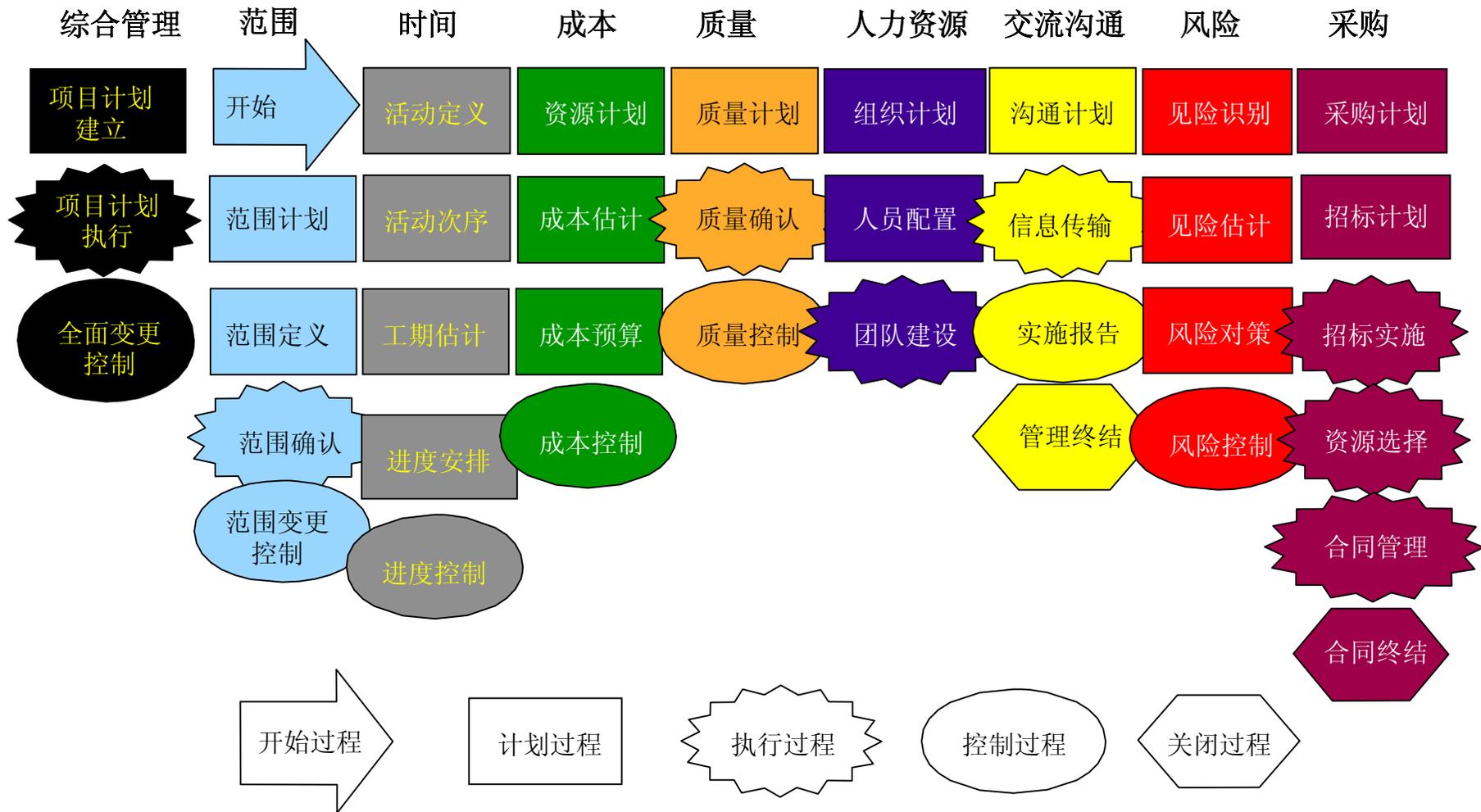
影响研发项目成功的因素

- 项目的目标、范围是否明确；
- 是否获得领导的积极支持；
- 项目的组织是否健全、稳定；
- 是否建立了有序的、有效的、良好的沟通渠道；
- 是否具有有效、全面的项目管理，严格的变更控制；
- 是否建立了良好的、积极的、团队合作的工作氛围；
- 项目经理PM的经验。

研发项目失败的主要因素

- 项目目标不明确
- 缺乏有力的领导
- 缺乏高层管理者的支持
- 技术问题没有解决
- 不合理的预测
- 跨部门协作不得力
- 计划和控制不力
- 过多的不可控变动
- 责、权、利不清
- 资源配备、供给欠佳
- 缺乏有效的沟通
- 项目经理缺乏魅力、影响力

现代项目管理知识体系的构成



项目模拟（组建团队）

选择项目

- 1、学员工作中的具体项目
- 2、世界屋脊---西藏行
- 3、我爱我家(家装)
- 4、花好月圆(广东)
- 5、初为人父母
- 确定项目目标：目标、时间、成本、质量。。。
- 确定项目初步的范围
- 确定项目可交付的成果
- 确定项目关键里程碑（MILESTONE）
- 项目需考虑的其它因素：制约、假设、风险。

团队名称、团队标志（logo）、团队口号

定义角色：项目经理（PM）

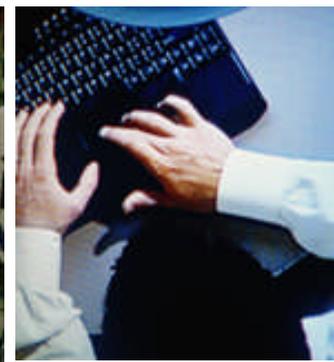
时间：15分钟

上台陈述（2分钟/团队）





DINGLI COMMUNICATIONS CORP.,LTD.



目录 *Content*

- 一、研发项目与项目管理的基本概念
- 二、研发管理、IPD、CMMI、敏捷的定义及关系**
- 三、软件开发模型与工作方法
- 四、研发项目管理及组织
- 五、研发项目计划制定
- 六、研发项目计划控制
- 七、在KD研发中心的某一天
- 八、研发项目精细化管理
- 九、KD研发中心项目管理体系

一. 产品研发管理的理念

- 企业研发管理的指导思想是：关注结果，重视过程。
 - “关注结果”：以最终产品获得经济效益来衡量研发业绩，追求利益最大化
 - “重视过程”：将期望的成果分解到每个过程域（即工作环节）去实现。
- 衡量研发工作优劣的三个关键指标是：质量、进度和成本。
- 企业研发管理的基本目标：让所有人员有条不紊地开展工作，在预定的时间和成本之内，开发完成质量合格的产品，从而使企业和个人获得预定的利益。
- 企业研发管理的奋斗目标：调动一切积极因素，努力提高产品质量、提高工作效率并且降低成本，使企业和个人获得比预定目标更多的利益。
- 企业里大部分工作是成熟的，有现成的模式可以套用，这类工作应当靠流程制度来管理，可比喻为“法治”。企业中还有一部分工作可能是独特的，并不适宜套用流程制度（也可能没有流程制度可以套用），相关人员要当机立断、高效地处理问题，可比喻为“人治”。国内大部分IT企业的研发管理现状是：“法治”太少，混乱的“人治”太多。

产品开发之道：正确地开发产品

1 概念

- 基本要求：项目团队在预定的时间和成本之内，开发完成合格的产品；
- 努力方向：项目团队尽最大努力把产品做得好、做得快并且少花钱。
- “质量、进度、成本”通常是衡量产品开发过程优劣的三个关键指标。一般说来，质量、进度、成本之间存在对抗关系。根据常识可知，要想同时提高产品质量、进度并且降低开发成本是非常不容易的。

2 关于质量

- 对于一个特定的产品而言，我们首先要判断什么是它的质量要素，才能给出提高质量的具体措施，而不是一股脑地想把所有的质量属性都做好，否则不仅做不好，还可能得不偿失。简而言之，能成为卖点的质量属性才是质量要素，才值得开发人员关注。
- 提高软件质量的最终目的为了获取尽可能多的利润，而不是出于对完美质量的追求；如果某些质量属性并不能产生显著的经济效益，我们可以忽略它们，把精力用在经济效益贡献最大的质量要素上。
- 企业必须权衡质量、进度和成本，产品质量太低了或者太高了，（可能）都不利于企业获取利润。企业理想的质量目标不是“零缺陷”，而是恰好让广大用户满意，并且将提高质量所付出的代价控制在预算之内。

产品开发之道：正确地开发产品

3 关于进度

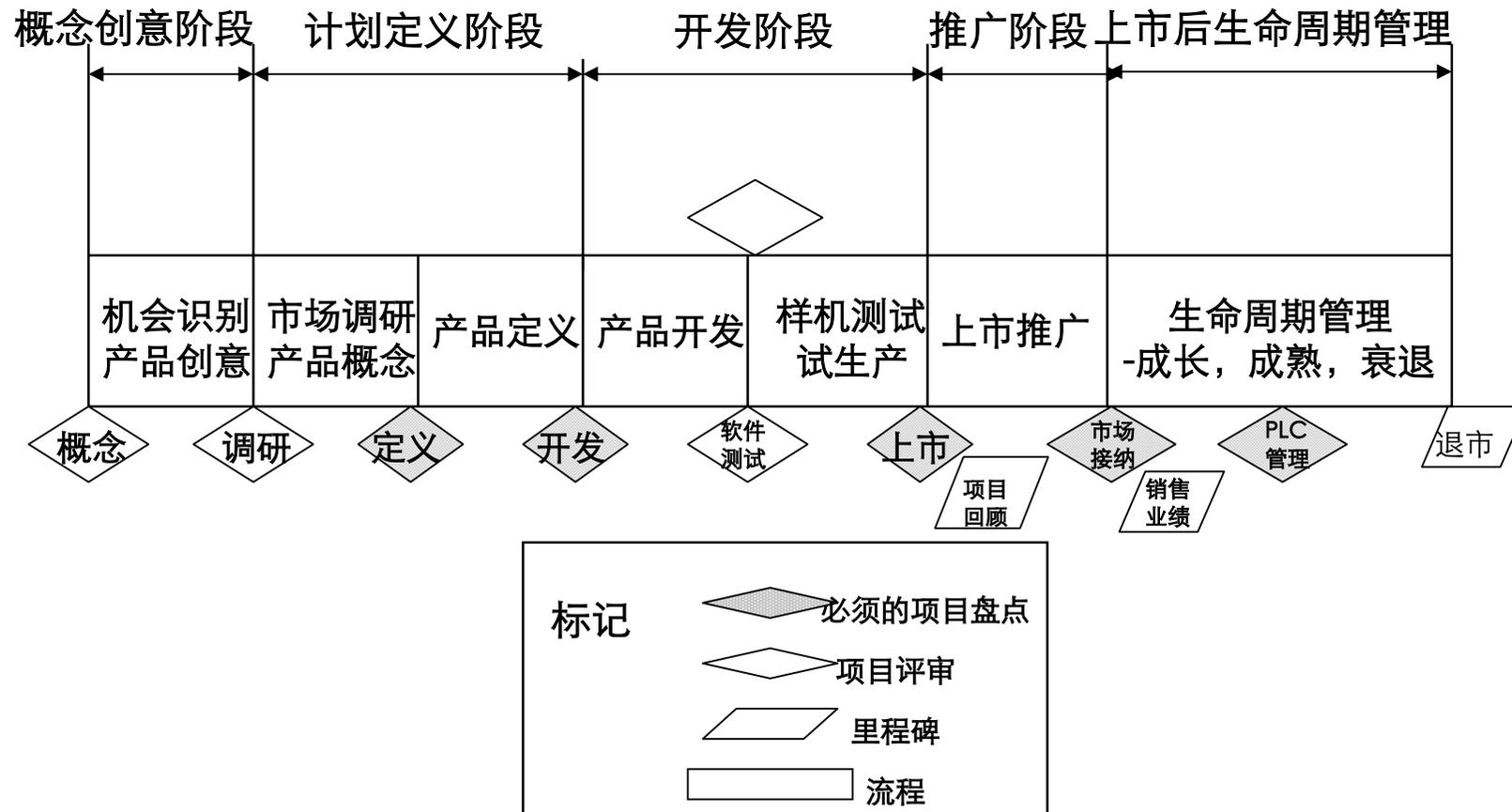
- 提高工作效率的前提条件是所有工作成果的质量必须合格。否则，工作效率越高，软件中的缺陷就越多，那么用于测试和维护的代价也越高，得不偿失。所以提高质量、提高效率都不是喊口号，要根据企业的目标和当前实力，量力而行。
- 在不对质量和成本产生负面影响的前提下提高工作效率，这才是真本事，常见措施有：
 - 提高项目成员的工作技能。使他们在开发产品时不仅做得好而且做得快。
 - 制定合适的软件过程。软件过程定义了做事的主要步骤，如果过程混乱，做事颠三倒四的话，势必伤害生产率。
 - 提高复用程度。复用就是指“利用现成的东西”，软件中可以复用的对象有设计模式、代码库、文档模板等等。软件人员应当懂得复用别人留下的成熟可靠的成果（可能要花钱去买也可能是免费的），并且还要给自己留下可以在将来复用的东西。
 - 使用高效率的开发工具和管理工具。

产品开发之道：正确地开发产品

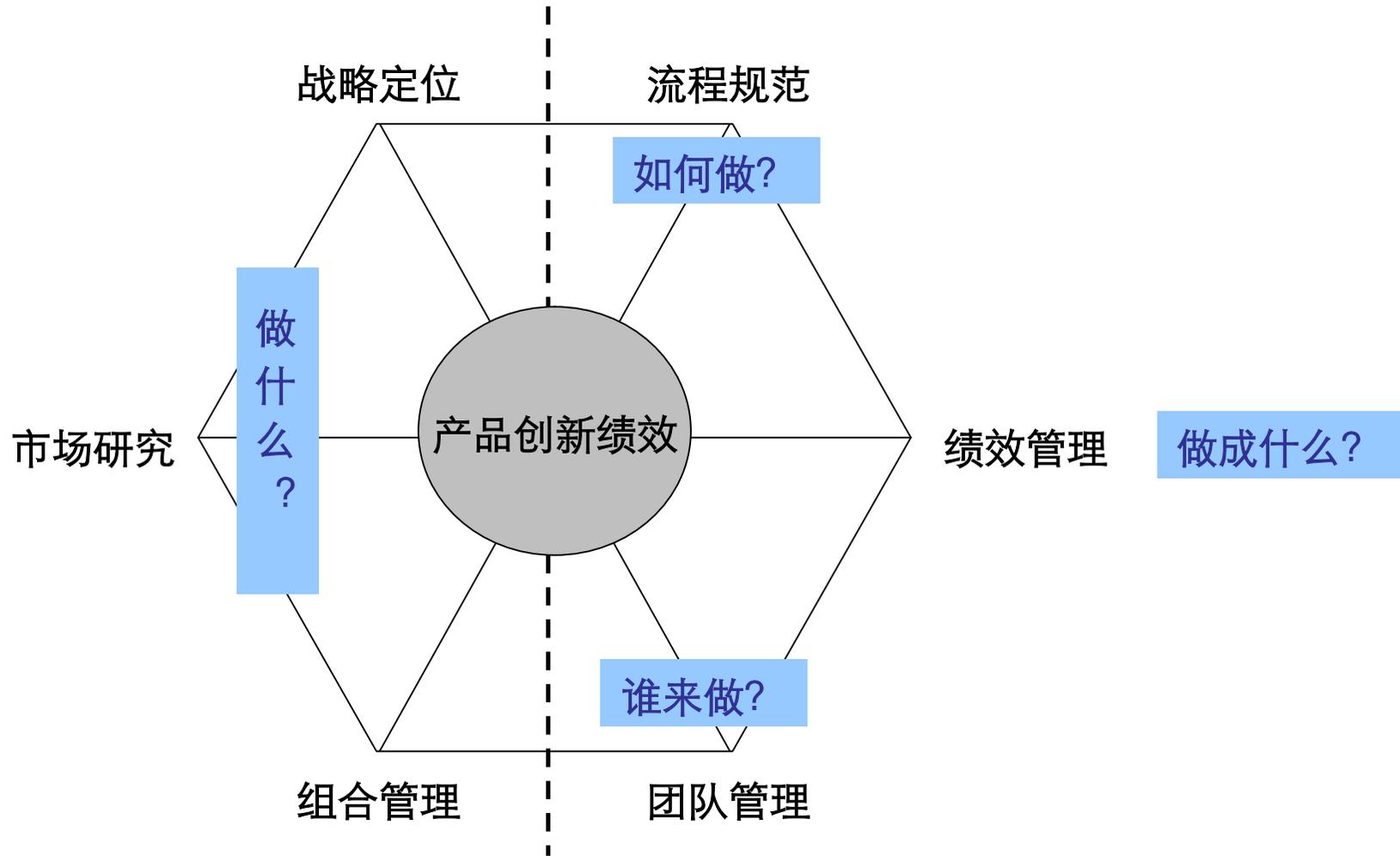
4 关于成本

- 一般地，开发成本和维护成本是软件的主要成本构成。
- 除了软硬件基础设施的成本外，人力资源成本占了开发成本的主要比例。
- 人们常常关注开发成本而忽视了维护成本。软件刚卖出去的时候，销售价格肯定比开发成本高，看起来是赚了钱。但是如果软件质量比较差的话，那么维护成本将是个无底洞，完全有可能把先期赚的钱给消耗光。所以人们不可为了压缩开发成本而放弃软件测试、技术评审等质量检查活动。前期偷懒将使后期遭殃。
- 在绝大多数情况下，设法降低成本将有益于企业获取更多的利润，但并不是绝对的。企业有短期目标和长期目标之分，为了使企业利益最大化，在某些时候企业会不惜成本地去抢占市场，以使未来获取更多的利润。产品的决策者一定要搞清楚质量、进度、成本之间的复杂关系，判断孰重孰轻，给出优化和折衷的措施。

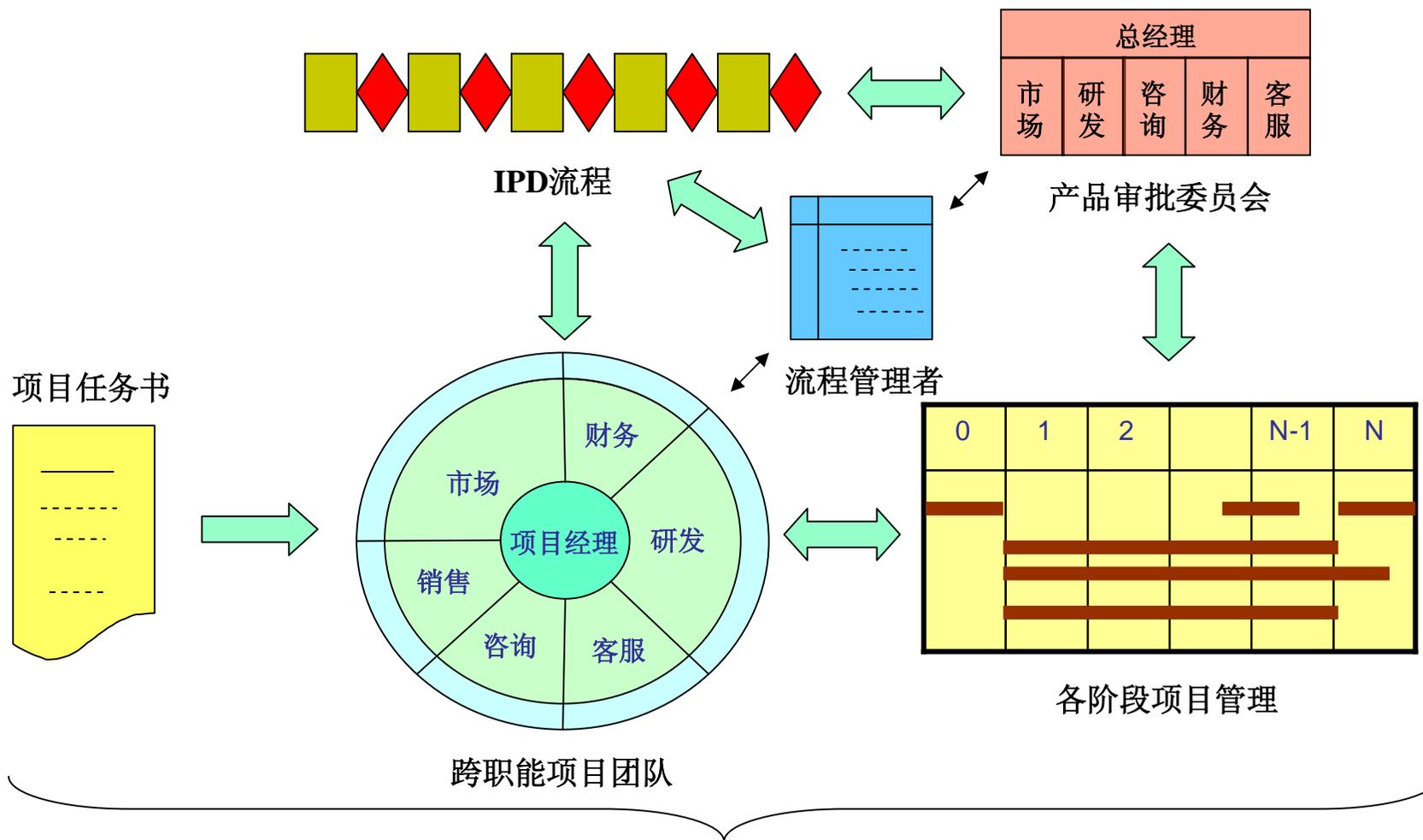
典型的生命周期(典型的研发产品开发流程)



研发产品创新六大关键成功要素（CSF）



研发产品开发管理框架



IPD绩效管理——“多、快、好、省”地开发新产品

二、集成产品开发流程(IPD)的概念

80年代开始，并行工程开始得到应用，通过改进产品开发流程，使产品开发的早期阶段能及早考虑下游的各种因素，达到缩短产品开发周期，提高产品质量，降低产品开发成本，从而增强企业竞争能力的目标。

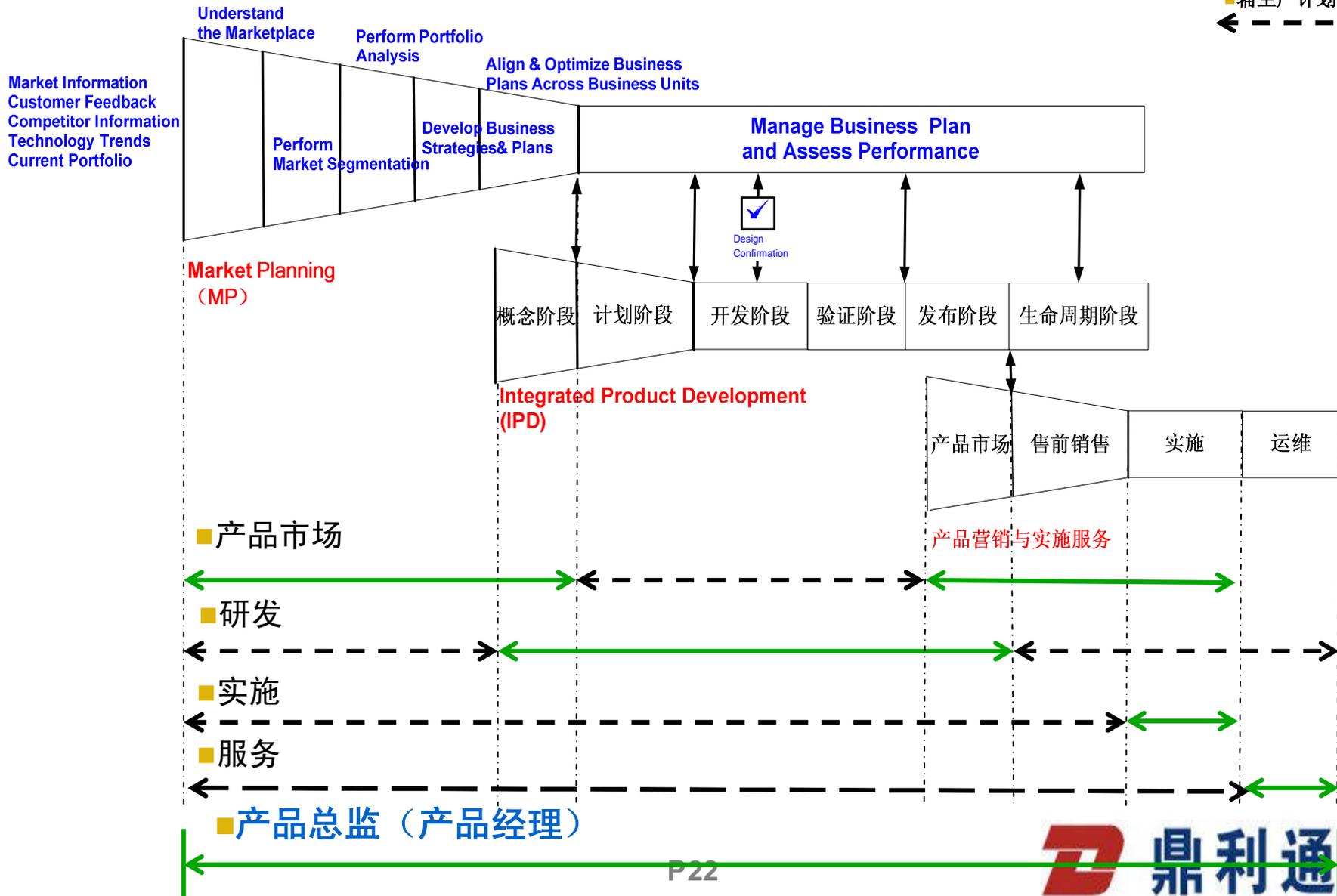
高效产品开发流程的精髓，强调产品开发不仅是研发人员的职责，而是市场、研发、工程、制造、客服和投资分析等相关部门协作的跨部门流程，产品的竞争力体现在集成产品开发流程的管理上。

IPD核心：

- 1、市场驱动的研发
- 2、按产品做计划而不是按部门做计划
- 3、对顾客负责而不是对老板负责，对事负责而不是对人负责

IPD中的定位

Integrated Portfolio Management Team (IPMT)



三、CMM 概述

CMM—Capability Mature Model能力成熟度模型

CMM的起源—美国卡内基梅隆大学软件研究（SEI）

Software Engineering Institute Carnegie Mellon University USA

它综合了以下几方面:

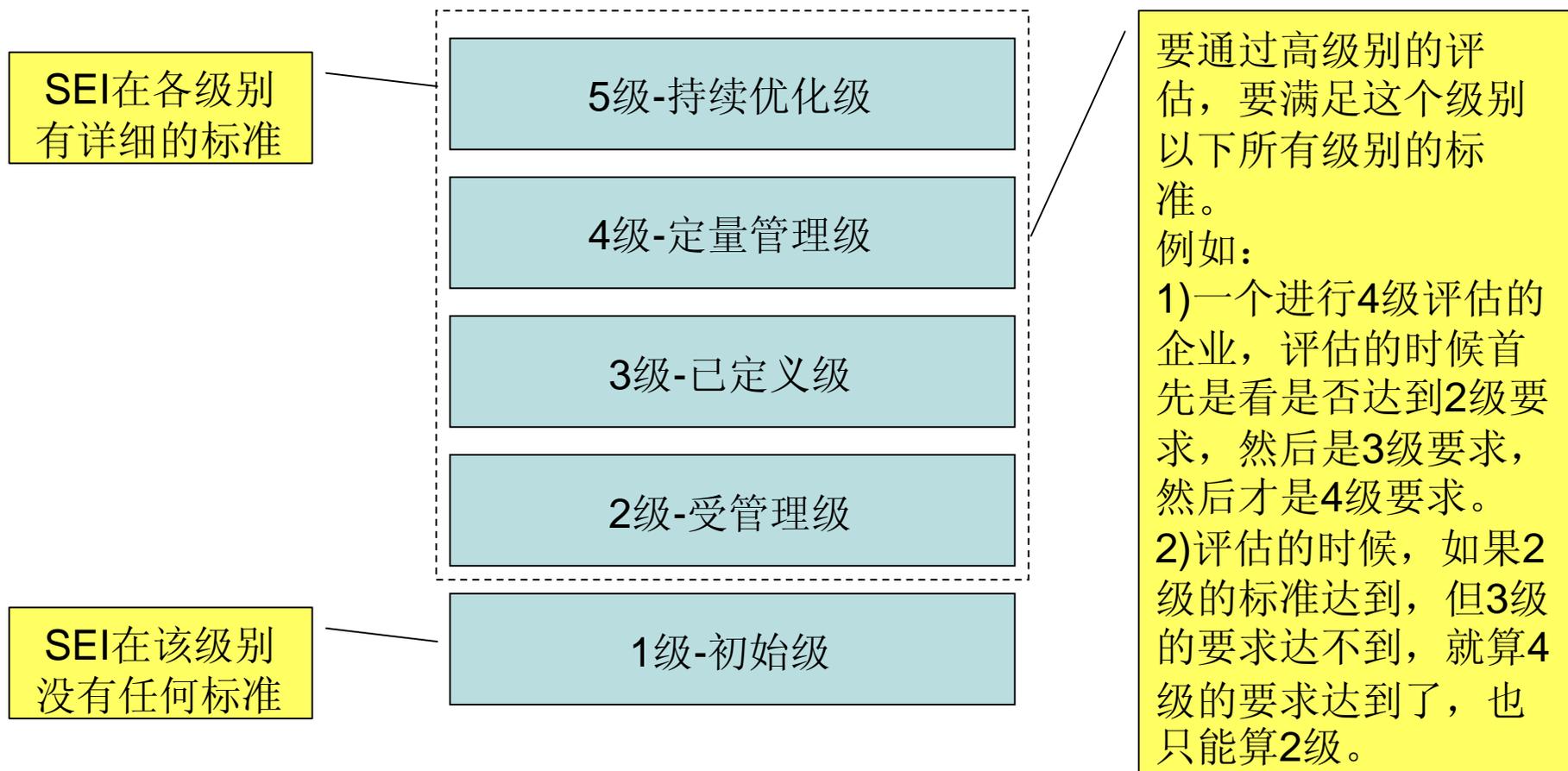
- System engineering
- Software engineering
- Integrated Product and Process Development
- Supplier Sourcing

该模型提供一套可供公众使用的准则，这些准则描述那些成功地实施了过程改进的组织特性；

该模型用“软件能力成熟度”来衡量这种软件综合能力；

她是一个如何做好软件的最佳实践的集合，已经得到全球实践证明，我们不必怀疑她的先进性。如果我们没有做好，那不是CMMI的问题，而是我们的理解与执行的问题。

CMMI的概念体系



实施CMMI的意义

- 能保证软件开发的质量与进度，能对“杂乱无章、无序管理”的项目开发过程进行规范。
- 有利于成本控制。因为质量有所保证，浪费在修改、解决客户的抱怨方面的成本会降低很多。现在绝大多数情况是缺少规范制度，只是求快。项目完成后，要花很多时间修修补补，费用很容易失控。
- 有助于提高软件开发者的职业素养。每一个具体参与其中的员工，无论是项目经理，还是工程师，甚至一些高层管理人的做事方法逐渐变得标准化、规范化。
- 能够解决人员流动所带来的问题。公司通过过程改进，建立了财富库以共享经验，而不是单纯依靠某些人员。
- 有利于提升公司和员工绩效管理水平和持续改进效益。通过度量和分析开发过程和产品，建立公司的效率指标。

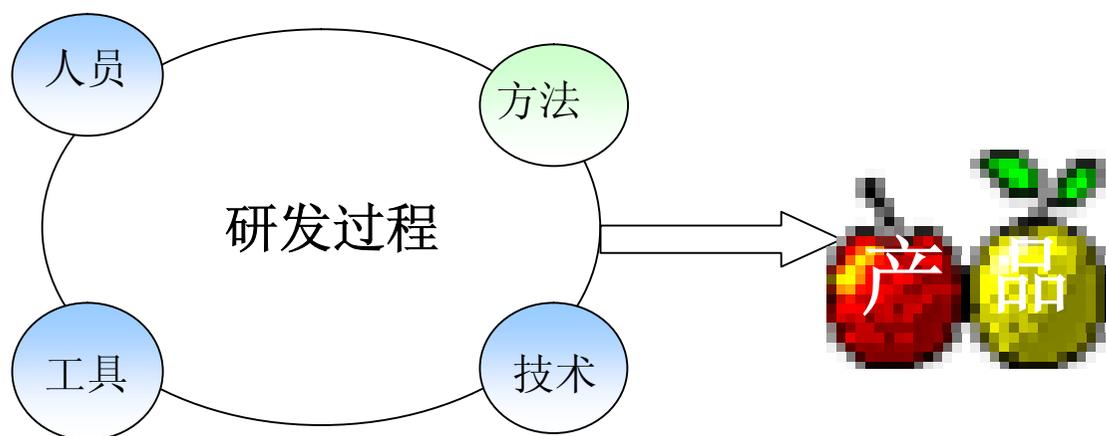
如何应用CMMI

- 应当根据企业的实际情况，既要裁剪CMMI过程域和实践，又要补充CMMI没有涉及的过程域和实践。企业领导和软件过程改进工作者必须明白：企业需要吻合商业目标、容易执行的软件过程规范。
- 什么是裁剪？
 - 裁剪不是指用剪刀把CMMI厚厚的书剪成薄薄的书，裁剪是要动脑筋的：要分析企业的业务特征，根据自身的人力和财力，选取CMMI文本中一些重要的东西，舍弃其它不重要的东西。至于什么是“重要的东西”，则要根据它对企业的贡献多少来衡量。
- CMMI都560页厚了，为什么还要补充过程域和实践？
 - CMMI对于软件开发和管理过程的论述非常深入，但是却没有涉及“商务过程”，例如没有谈立项管理、售前服务、售后服务等。这是CMMI很大的缺陷。
 - 企业开发产品的最终目的是卖出产品，赚取利润。如果软件过程规范中不考虑商务过程的话，会导致开发团队“闭门造车”，很可能开发出“技术上很好的产品，但却是商业上失败的产品”。

四、研发过程改进的概念

什么是研发过程？

- 人们使用合适的方法、技术、工具才能开发出用户需要的产品。过程是指“人，方法，技术和工具”的集合。
- 过程被写成文档后，变成了公司的“流程制度”，公司成员们依据“流程制度”开展工作，这叫“法治管理”。



研发过程改进的概念

研发过程与产品有什么关系？为什么要重视过程？

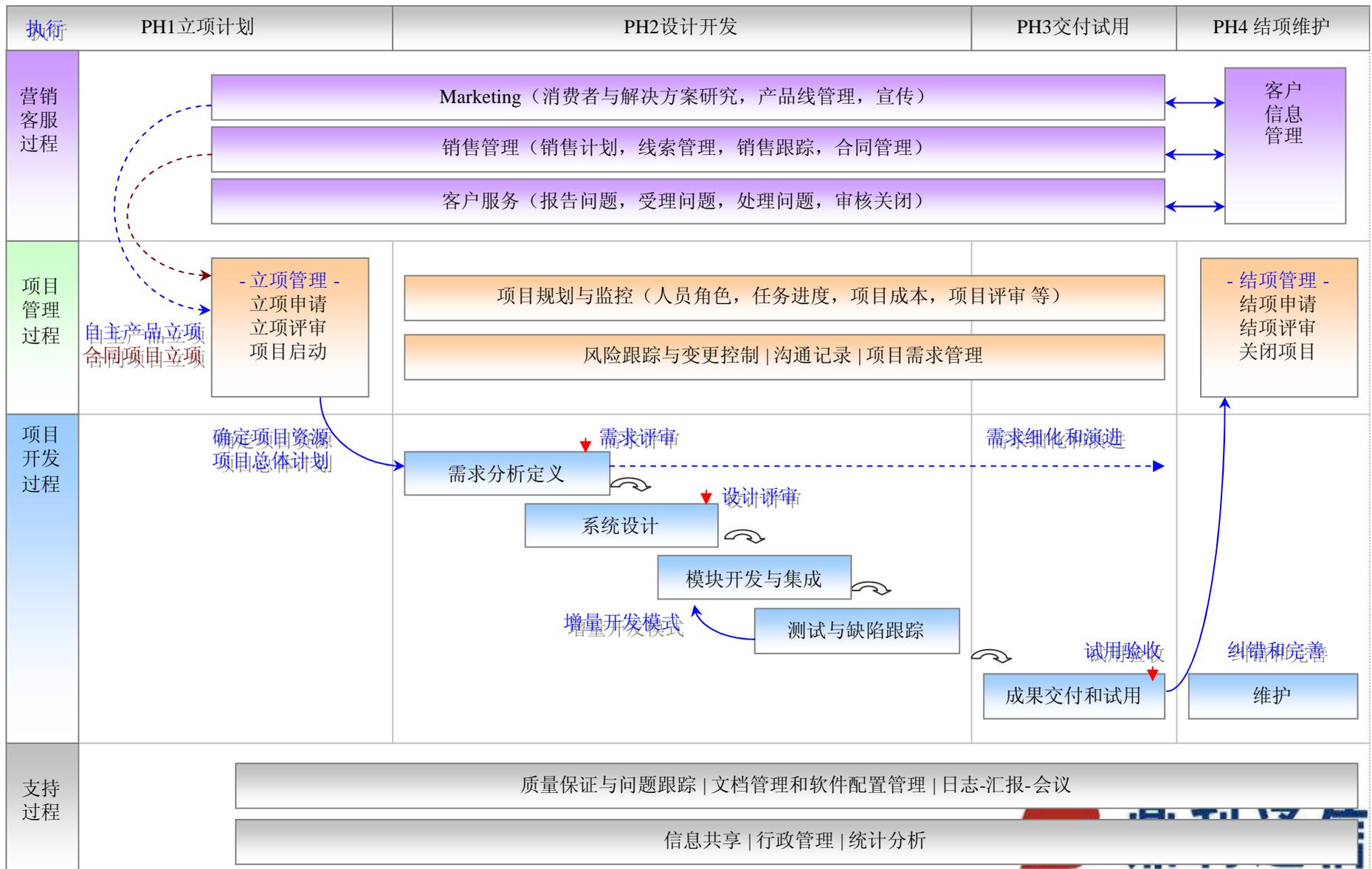
- 软件产品不能靠人们的意念瞬间完成，它需要一个研发过程。一般情况下，好的过程才可能得到好的产品，而差的过程会得到差的产品。
- 当然也有相反的情况，有些人在混乱的过程中创造了很好的产品，也有些人在严谨的过程中生产出商业上失败的产品。但这类现象不具有指导意义。
- 为什么要重视过程？
 - 由于公司销售的是产品而非过程，人们常常只把眼光盯在产品上，而忘了过程的重要性。
 - 例如，领导对员工们下达命令时经常强调：“我不管你们怎么做，只要时间一到你们交付产品就行。”其实这是一句因果关系颠倒了的话，却在业界普遍存在。
 - 如果领导不关心员工怎么做（即做事的过程），往往会得到失望的结果。

软件过程改进和CMMI之间的关系

- 在二十世纪七、八十年代，软件工程的研究重点是需求分析、软件设计、编程、测试、维护等领域的方法、技术和工具，我们称之为经典软件工程。
- 应该说现代的软件技术、软件工具要比几十年前好不知道多少倍，可是如今绝大多数软件项目依然面临着质量低下、进度延误、费用超支这些老问题。人们逐渐意识到，由于机构管理软件过程的能力比较弱，常常导致项目处于混乱状态，过程混乱使得新技术、新工具的优势难以体现。经典的软件工程不是不好，而是不够用。
- 提高软件过程能力的实践通称为软件过程改进（Software Process Improvement）。软件过程改进的目的是：提高软件质量、提高生产率并且降低开发成本。从二十世纪九十年代至今，软件过程改进成为软件工程学科的一个主流研究方向，其中CMM/CMMI是该领域举世瞩目的重大成果。CMM/CMMI是世界范围内用于衡量软件过程能力的标准。

基于CMMI的集成化研发过程改进模型

企业战略管理（设计盈利模式 + 提升核心竞争力 + 优化组织结构 + 优化流程制度 + 企业文化建设）



五、CMMI-IDP-RDMS的对应关系

IDP的过程域		CMMI 2-3级过程域	RDMS功能模块
营销过程	产品构思和调研	无直接对应过程域	
	产品体验和宣传销售	无直接对应过程域	<ul style="list-style-type: none"> ✓Portal研发管理门户 ✓InfoShare信息共享系统 ✓Satisfy客户管理系统
	合同项目销售	2级-供应商协议管理	
	客户沟通和合同验收		
项目管理过程	立项管理和结项管理	3级-集成化项目管理	✓Future立项与结项
	项目规划与监控	2级-项目规划 2级-项目监控	✓Future项目规划与监控
	风险跟踪和变更控制	3级-风险管理	✓Future风险跟踪和变更控制
项目开发过程	需求开发与管理	2级-需求管理 3级-需求开发	✓Future需求管理
	软件系统设计	3级-技术方案 3级-产品集成	无直接对应功能
	模块开发与集成		
	软硬件系统集成		
	测试与改错	3级-验证 3级-确认	<ul style="list-style-type: none"> ✓Future测试管理, 技术评审 ✓Future发布管理, 缺陷跟踪
	部署试用		
	软件维护		✓Future维护记录
支持过程	软件配置管理	2级-配置管理	✓DocCenter文档管理, CVS等
	软件质量管理	2级-过程和产品质量保证	✓Future质量管理
	客户服务管理	无直接对应过程域	✓Satisfy客户管理系统
	统计分析	2级-度量分析	✓Analysis统计分析工具

六、CMMI vs 敏捷(什么是尽善尽美的过程)

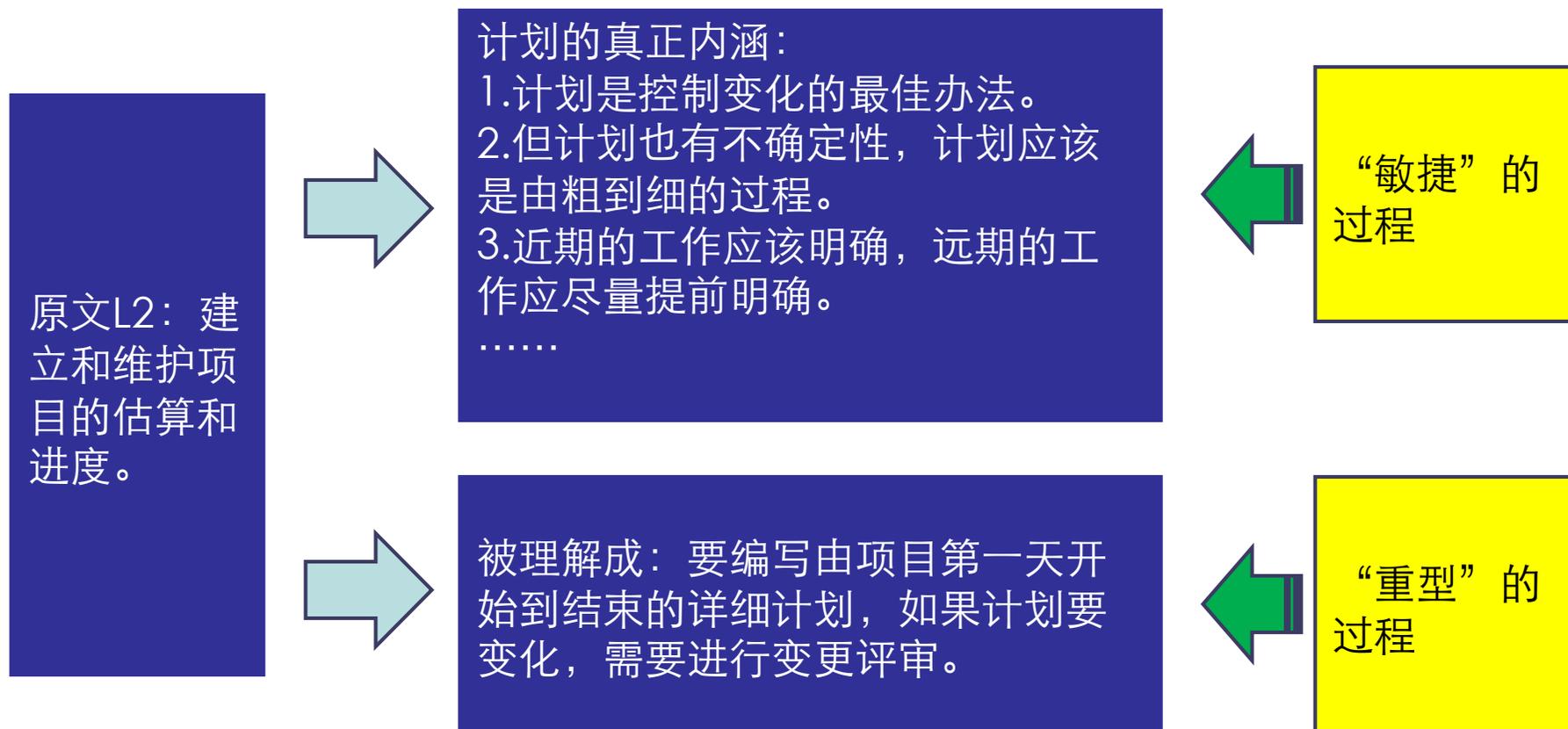
有人曾经这样问我：

- 当我们把过程做得尽善尽美，结果项目却失败了我们应该如何看待呢？
- 保证了过程却没有好的产品是不是对质量管理权威有所动摇呢？

选择题：什么是尽善尽美的过程？

- A、过程文档数量庞大。
- B、过程步骤多，评审多，需要写的文档多。
- C、满足CMMI5的要求。
- D、完美的过程应该经得起实践考验，能提高项目成功的概率！

CMMI的原意被曲解示意图



什么是敏捷过程？

过程的目的是为了让项目做得更好，而不是把项目“框”死。

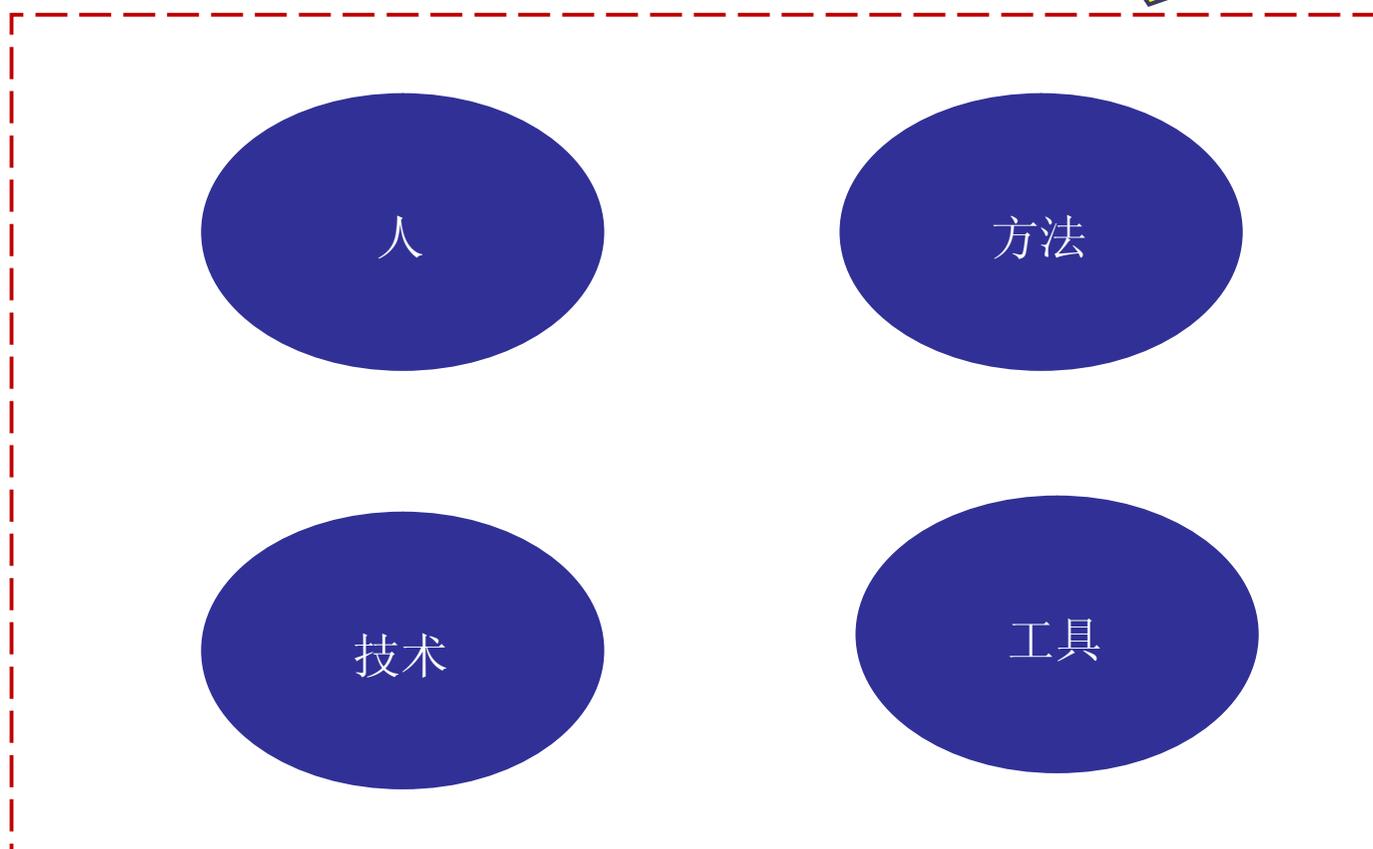
敏捷的核心思想，就在此！

敏捷过程

- 一个项目目标明确的过程。
- 有利于实现项目目标的事情，一定要做。
- 对项目目标没有帮助的事情，一律不做。
- 有效和高效是最重要项目管理原则。
- 敏捷的过程是让人愉快、工作起来有战斗力的过程。

什么是过程？

这4个东西是过程4要素，
而过程是这4者的有机结合！



还CMMI本色，让过程敏捷起来！

人

- 招聘
- 培训
- 激励制度

方法

- MSF
- RUP
- 敏捷开发
- XP开发
-

还CMMI本色，让过程敏捷起来！

技术

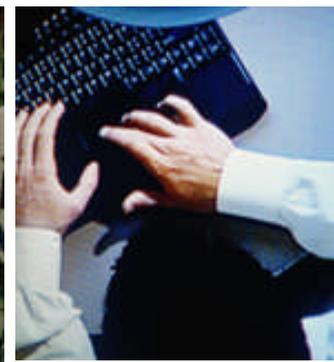
- 定好公司的技术路线
- 规划整个公司的技术学习
- 技术重用
- UML
-

工具

- SharePoint
- Excel
- Project, Project Server
-



DINGLI COMMUNICATIONS CORP.,LTD.



目录 *Content*

- 一、研发项目与项目管理的基本概念
- 二、研发管理、IPD、CMMI、敏捷的定义及关系
- 三、软件开发模型与工作方法**
- 四、研发项目管理及组织
- 五、研发项目计划制定
- 六、研发项目计划控制
- 七、在KD研发中心的某一天
- 八、研发项目精细化管理
- 九、KD研发中心项目管理体系

软件开发流程模型

软件开发流程定义了谁、什么时候、怎样、做了什么

- Who – 团队的哪些成员
- What – 采取什么行动
- When – 相对其他行动，本行动何时完成
- How – 行动的细节和步骤

软件开发流程模型

- 瀑布开发模型（生命周期法）
- 螺旋开发模型
- 演进/敏捷开发模型（循环往复法）

瀑布开发模型

瀑布模型，软件开发行动按照一定次序进行，每一步都是基于前一步的行为

- 认识到不同阶段之间的反馈相互关联，因此承认会有诸如设计影响需求，系统编码引起设计的变更等情况的发生
- 开发原型系统和需求分析、设计行动并行进行

缺点

- 此模型是固定、严格的开发方式的代表
- 需求在项目的生命周期中被“冻结”
- 开发流程自行发展
- 范围管理的具有相当的挑战性
- 如果项目超出范围（与原来设定的范围相比）没有交付的成果，混乱，质量差，软件废弃

螺旋开发模型

螺旋模型：开始阶段开发一系列风险驱动的原型，然后采用结构化的类似瀑布模型的流程

- 提供可感知的指引（roadmap）帮助处理某些需求方面的挑战
- 可提供给用户和客户多个反馈的机会

缺点：

- 对于超出边界的项目，没有更好的效果，尽管一两个原型具备可操作性；

演进/敏捷开发模型

演进/敏捷模型或者反复迭代方法实际是瀑布模型和螺旋模型的综合，降低生命周期各阶段与各阶段进行的软件开发活动之间的耦合性

- 允许我们在各迭代阶段，重新访问各种活动，比如需求、设计、实现
- 每一次迭代都计划降低在开发活动已经表现出来的风险

产品生命周期的阶段

初始期

- 主要精力在于理解商业需求，项目边界，实施方案的可行性
- 进行问题分析；创建Vision文档；评估进度安排和确定预算；确定项目风险因素

加工期

- 完成系统需求定义；确定初步可执行的构架；前期可行性原型开发和展示

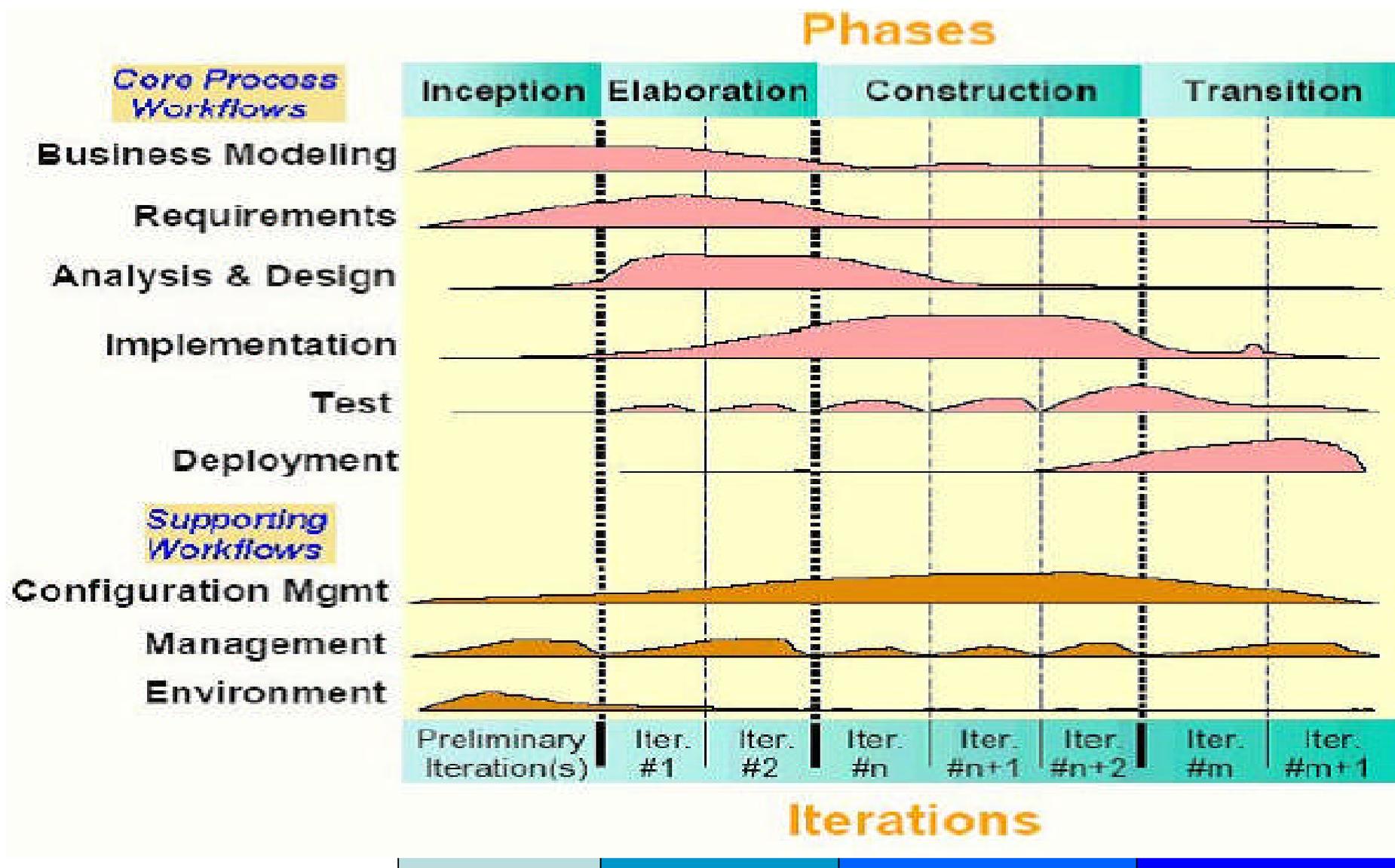
建设期

- 主要任务是实施
- 绝大部分编码工作完成；整体构架和设计全部完成

过渡期

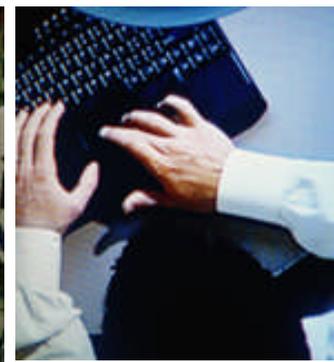
- 完成Beta测试；系统用户和维护人员接受培训
- 将通过测试的应用移交给用户团体，部署待用

演进/敏捷开发模型的工作流





DINGLI COMMUNICATIONS CORP.,LTD.



目录 *Content*

- 一、研发项目与项目管理的基本概念
- 二、研发管理、IPD、CMMI、敏捷的定义及关系
- 三、软件开发模型与工作方法
- 四、研发项目管理及组织**
- 五、研发项目计划制定
- 六、研发项目计划控制
- 七、在KD研发中心的某一天
- 八、研发项目精细化管理
- 九、KD研发中心项目管理体系

项目调研

在进行产品预研和立项之前应该做好充分的调研工作，调研内容包括：

- 市场可行性调研
- 技术可行性调研
- 经济及成本可行性调研
- 知识产权调研

调研过程中，应该充分利用资源共享的平台，与各职能部门及分支机构进行信息的交流，同时要咨询相关专家的意见，而不要单单利用个人的力量去闯荡。

项目预研

- 何时进行预研

当该产品或项目在市场前景不明确、技术难度较大且暂无良好解决方案，该产品生产困难或较难发挥公司总体研发能力，但该项目有可能成为市场新的增长点或与公司战略相符，这时该项目可进入预研阶段。

- 预研的目的

通过预研对整个产品在开发、生产、营销等各个环节（全流程）上运作做出评估或对存在的技术/非技术难点寻找解决方法，以减少公司开发该项目的风险。

项目预研

- 预研过程

项目进入预研后，由职能部门根据预研项目的性质、难度、相关人员的技术背景、目前工作安排等各种因素，草拟预研组人员名单并报请相关组织批准。然后职能部门根据市场情况、人力配备、待预研解决的问题性质等因素制订预研的“目标计划”，并以“预研任务书”形式正式备案。预研组根据任务书进行多方面的跟踪研究，并取得阶段性成果，最后由相关组织（专家组）对阶段成果进行评审，决定是否转立项或继续预研。

需求：客户并不总是知道自己想要什么

- 他们知道他们不想要什么
- 当产品市场推介到一定程度，产品有了雏形之后，客户可能会意识到新的需求
- 看到样机时，客户知道他们想要什么
- 要认真地考虑客户需求---- 如果一个产品包没有被目标客户所使用、使用不够或错误地使用，则该产品可被视为一个失败的产品。

《预研项目建议书》模板示例

1 综述

- 1.1 预研项目提案
- 1.2 预研项目风险
- 1.3 建议

2 项目背景和必要性

- 2.1 项目背景
- 2.2 项目必要性
- 2.3 财经可行性

3 预研目标

- 3.1 预研目标、主要技术指标和参数
- 3.2 预研产品规格及性能

4 技术可行性

- 4.1 项目的基本内容
- 4.2 公司内可利用的相关技术资源
- 4.3 关键技术及解决方案
- 4.4 阶段评审、测试验收标准和方法
- 4.5 技术风险分析及防范措施

5 知识产权可行性

- 5.1 国内外技术及标准跟踪研究
- 5.2 专利技术情报的全面检索与充分利用
- 5.3 关键技术知识产权保护策略

6 市场可行性

7 经济可行性

8 预研项目计划

- 8.1 项目工作任务和时间安排
- 8.2 项目所需的人、财、物、信息等资源
- 8.3 项目总预算和分阶段预算
- 8.4 阶段划分、评审及测试验收
- 8.5 计划风险分析及防范措

9 其它

项目的组织形式

- 职能型组织结构

不注重客户，人们强烈忠诚于自己的部门，而不是项目或客户；但减少了重复工作，有专业化的好处。

- 项目型组织结构

成本低效，项目间缺乏知识信息交流；但能控制资源，对客户高度负责。

- 矩阵式组织结构

矩阵式组织结构

主要特征：

- 组织中存在“两个上级”系统
- 双重评价系统与控制系统
- 各层次人员各有其独特的重要作用
- 职能部门往往有双重领导责任

矩阵式组织结构

优点：

- 有利于加强责任制
- 有利于充分利用技术部门的人力与物力
- 有利于员工的专业增长
- 有利于扩展员工的知识面与眼界
- 易于适应工作任务与客观需要的变化
- 有助于效益的提高
- 有利于高层领导集中精力于全面性、长远性、战略性的决策

矩阵式组织结构

缺点与困难：

- 为保持两个系统之间的平衡，需要管理上进行不断的调整
- 需要在管理上不懈努力，以达到较高的经济效益
- 双重报告制度引起矛盾和混乱
- 渠道的增生会造成信息的阻塞
- 责任的重叠会引起竞相争夺地盘和使考核困难
- 由于距离远、语言、时间、文化等相去太远，这种混乱与矛盾也不易处理和解决

组建项目组

一个实力强大、团队合作的项目组是项目的核心，也是项目成功的保障。作为项目经理，为了整个项目的成功，不仅要给予项目组成员技术指导及管理方面的专门知识，还得加上你无形而具深义的热情和支持。营造一个团结而宽松的工作氛围是项目经理的基本职责之一。因此，在组建项目组时你就得从以下方面考虑：

- 1、建立一个结构合理的项目组
- 2、寻找合适的人选，熟悉他们技术方面管理方面的长处和弱点，了解他们的能力
- 3、树立并保持项目组的团队精神
- 4、争取管理部门的支持
- 5、项目组内经常的有效的沟通

项目组职责

不同管理层次的跨部门团队对不同类型的决策负责：

划分

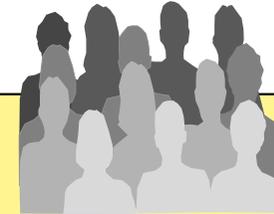
IPMT - 集成产品管理团队

硬件开发
硬件市场销售
供应

软件开发
软件市场销售
采购

财务
生产
制造

角色：管理区段组合，授权投资



PDT - 产品开发团队

硬件开发
硬件市场销售
供应

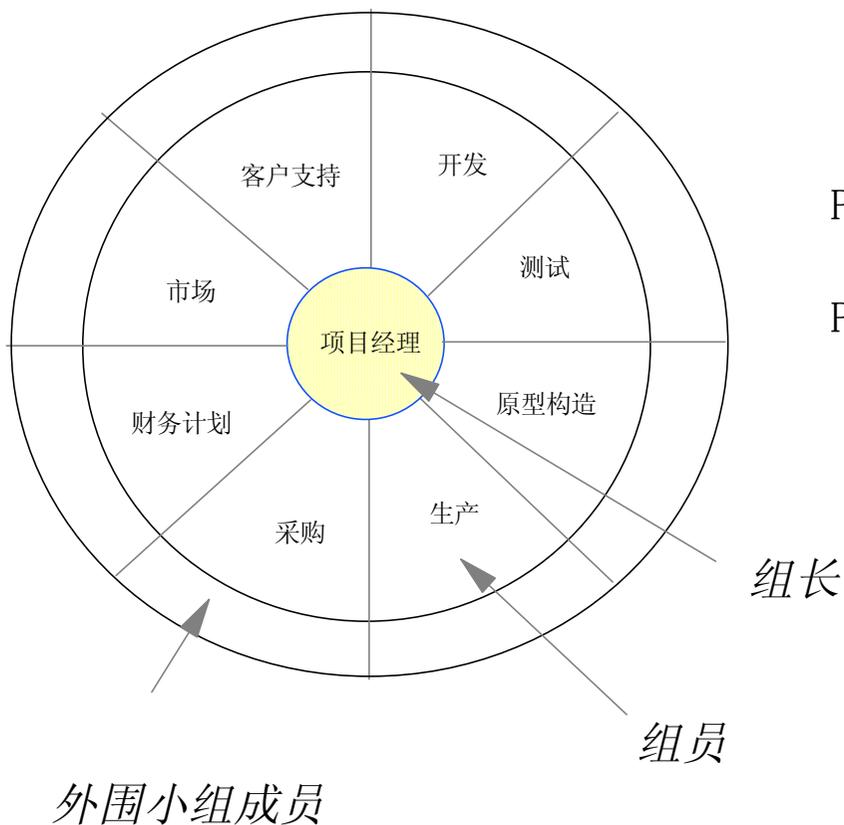
软件开发
软件市场销售
采购

财务
生产
制造

角色：开发提供的方案，执行合同

组建产品开发团队PDT

PDT的组织结构



PDT是临时小组

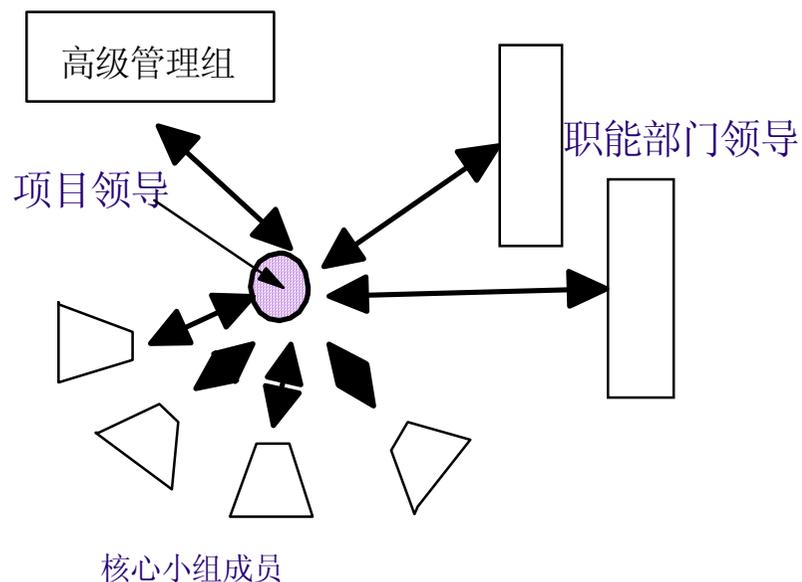
- 在项目开始时成立
- 在产品成功发布后解散

PDT成员在概念阶段一起作整个项目的计划

PDT成员在计划阶段一起管理整个项目



项目经理的角色及义务



领导项目组

- 指导产品从概念设计到市场接受
- 保证实现设计、收益、市场份额及利润目标
- 解决冲突

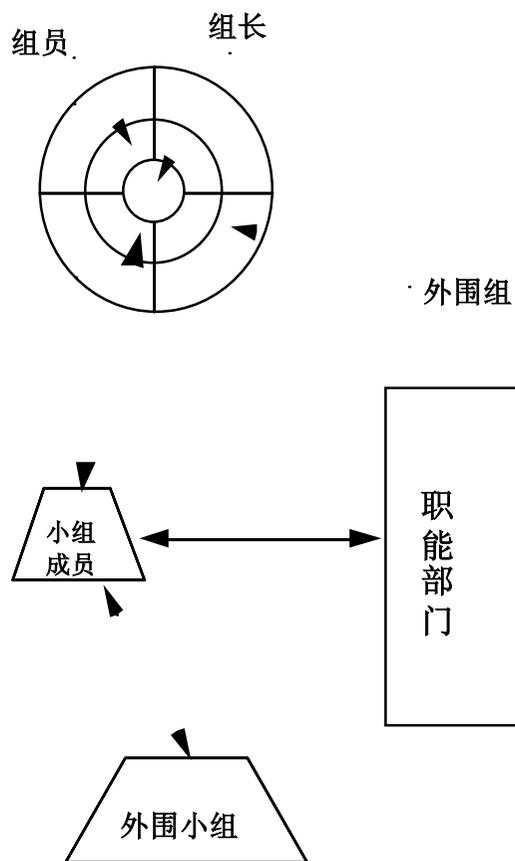
管理项目

- 制定项目计划及预算
- 确定/管理参与项目的人员/资源（与职能部门经理协调）
- 跟踪相对于项目基线的进展

与管理层沟通

- 提供项目进展状况
- 准备并确定决策评审点
- 作为产品领导
- 提供对项目组成员的工作绩效评审的输入

PDT小组成员的角色及义务



小组的职能专家

- 解决问题
- 在设计和项目决策时代表职能部门
- 共同负责小组的最终结果
- 对计划、预算、关键问题等的进展情况进行汇报
- 对功能部门的交付负责

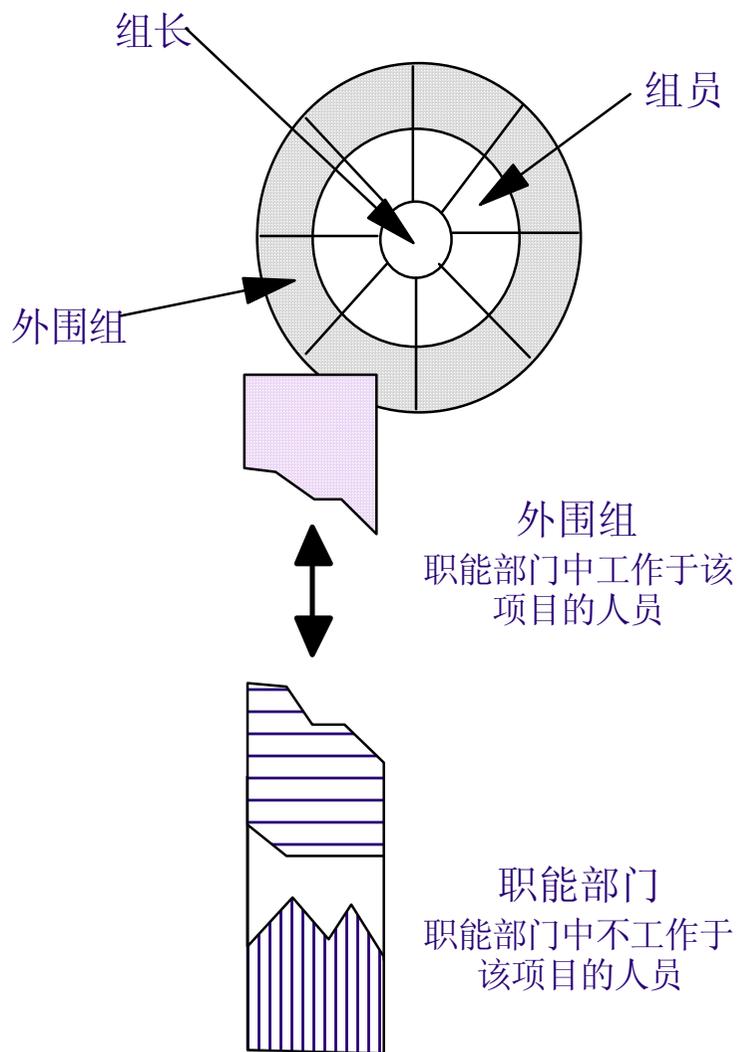
与职能部门的桥梁

- 向职能部门经理汇报项目情况
- 应用职能部门的策略、工具和标准

协同外围小组的活动

- 管理职能部门的项目计划和预算
- 负责PDT与职能部门间的信息交换
- 在职能部门内对设计/项目进行评审

外围小组成员的角色及义务



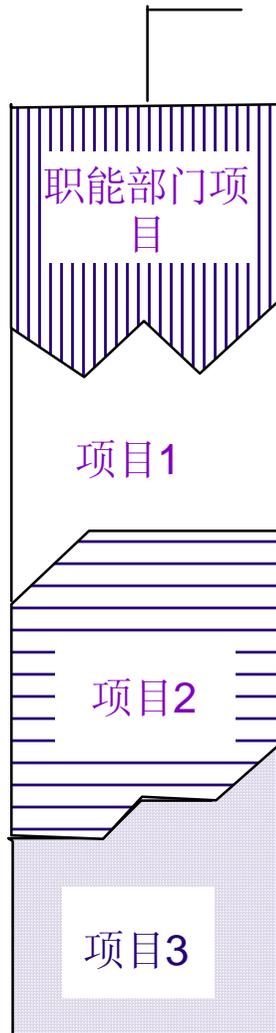
独立完成产品定义、市场交付、设计、测试等工作

- 关注于特定的功能性任务
- “Just do it”

在特殊情况下，PDT小组可能没有外围小组

- 非常小的项目
- 职能部门在项目中的工作不多

职能部门经理的角色及义务



提供技术领导

- 定义职能部门的策略、指导原则、工具和标准
- 协调跨项目的技术合作
- 制定并维护开发流程指导方针

发展并管理职能部门

- 建立优异的职能部门团队
- 执行职能部门预算
- 雇佣/解雇员工、培训员工及对员工进行绩效考评
- 领导职能部门项目

支持PDT工作

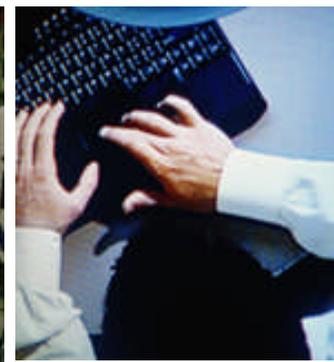
- 确定项目开发的人员及资源
- 参与设计及评审

培养项目经理所需要的能力

- 1、获取经验
- 2、寻求别人的反应
- 3、自我批评总结，改正错误
- 4、与一些具有你想学习的技能的项目经理进行探讨
- 5、参加培训项目
- 6、参加组织团体
- 7、阅读
- 8、参加自愿活动



DINGLI COMMUNICATIONS CORP.,LTD.



目录 *Content*

- 一、研发项目与项目管理的基本概念
- 二、研发管理、IPD、CMMI、敏捷的定义及关系
- 三、软件开发模型与工作方法
- 四、研发项目管理及组织
- 五、研发项目计划制定**
- 六、研发项目计划控制
- 七、在KD研发中心的某一天
- 八、研发项目精细化管理
- 九、KD研发中心项目管理体系

周密而严谨的计划 = 项目成功了一半!



项目目标--定义项目参数

- ✓ 定义项目参数
- ✓ 定义项目参数的关键问题
- ✓ 定义项目目标书（POS）
- ✓ 定义项目交付物及方法
- ✓ 定义项目参数的关键行动

定义项目参数的关键问题

- 项目范围是什么？
- 项目什么时候完成？
- 给项目分配什么资源？
- 是否有一份清晰的、简明的、字数少于25的项目目标书？
- 项目的主要交付物或者后果是什么？
是否很好地定义了主要的交付物？
- 每个主要交付物是否都有书面的是/否清单？
- 主要的交付物是否设定了目标完成的时间？

项目目标书（POS）

除了这三个参数（范围，进度，资源）以外，一份好的POS还包括几项其他重要性：

单词数不超过25（该限制迫使POS精确）

使用普通的语言，避免行话和首字母缩合词。

清楚，简洁。

理想的POS应是一种愿景，形成了挑战和一定程度的兴奋。

项目目标定义的SMART 原则

明确性 (Specific)

—— 最终目标是否明确？应该做到哪一步以及何时完成？

可度量性 (Measurable)

—— 能在多大程度上测量最终目标的完成情况？

可完成性 (Achievable)

—— 在规定时间内，最终目标是否合理，能够实现 Ref. attainable

相关性 (Relevant)

—— 最终目标是否很重要、很有价值，是否值得进行下去？ Ref. Realistic

可跟踪性 (Trackable)

—— 能够对整个项目进程进行跟踪检查吗？ Ref. Time-bound

定义项目的主要交付物

主要交付物是提炼完善了POS中对范围的定义。主要交付物是管理人员注意力焦点所在的主要项目的后果或者结果。

因为主要交付物是作为一种让管理人员将注意力集中在关键项目结果上的一种工具，有关这些交付物应该是什么以及多长时间应交付这些方面具体的指导方针很少。基本的“经验之谈”是：项目经理和团队应提前决定他们致力于哪些关键的有形结果。

项目计划的作用

1、计划是连通团体的经脉

- 压力自上而下充分传递
- 提高团队工作效率
- 明确职责

2、计划是走向目标的诺言

- 确定工作总目标
- 控制开发进程
- 计划是工作的指南针

3、计划是交流沟通的工具

- 工作得以量化
- 获得关键路径
- 合理地调配资源
- 清晰地反映产品状态信息

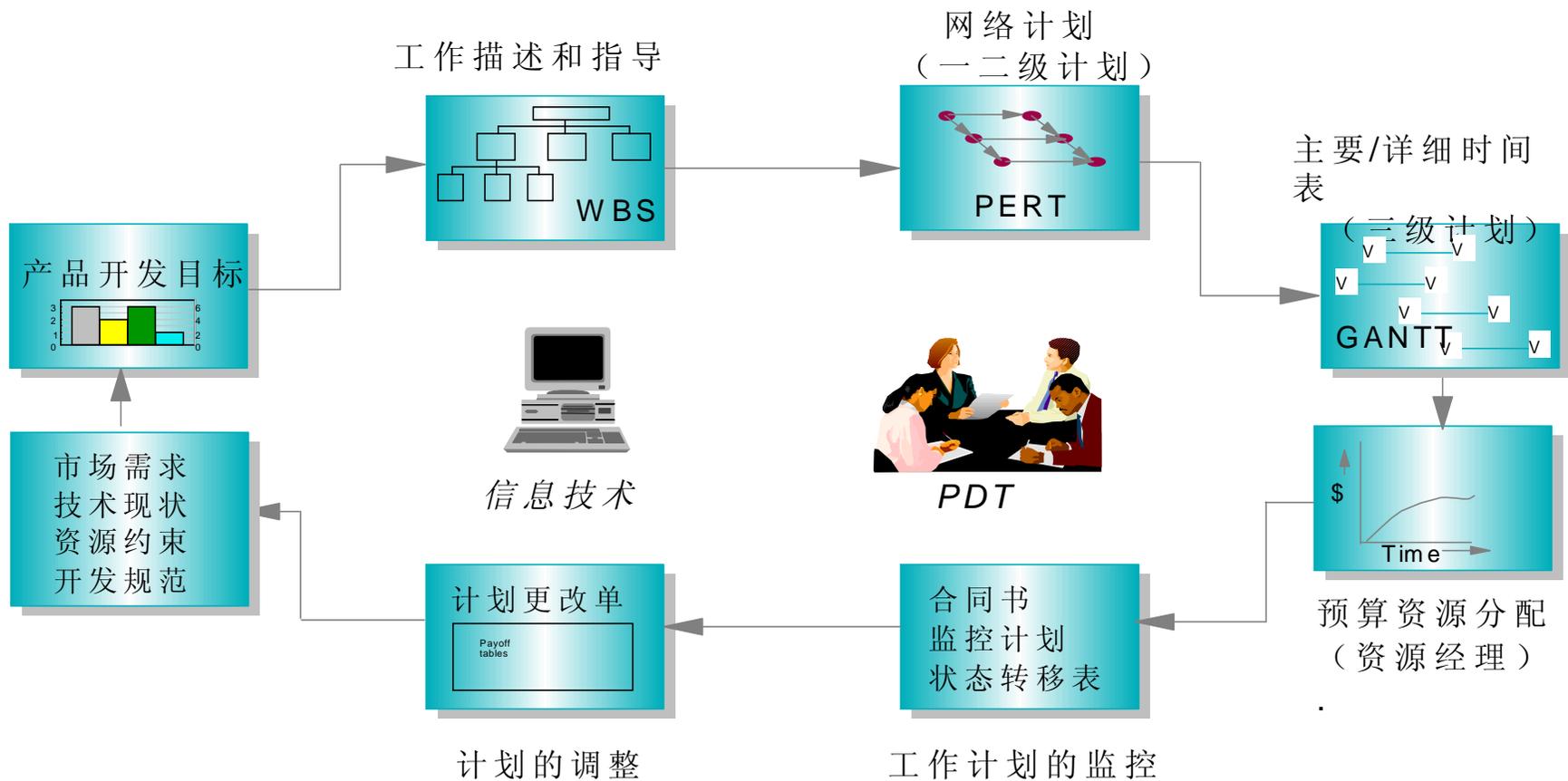
4、计划是实现成功的保证

- 规范开发活动
- 约束和协调的依据
- 问题的预警与防范

一个完整的计划包括

- 任务名称及层次
- 资源
- 成本
- 时间进度
- 完成标志
- 上层任务的约束
- 下层任务的配合
- 阶段里程碑

计划制订的流程



计划制定的要素

完整性:

- 是否包含了版本及所有特性的计划;
- 是否全流程的计划(硬件、软件、测试、制造、市场技术、技术支援、生产等);
- 是否产品卖出去的计划(资料、宣传、操作指导等);

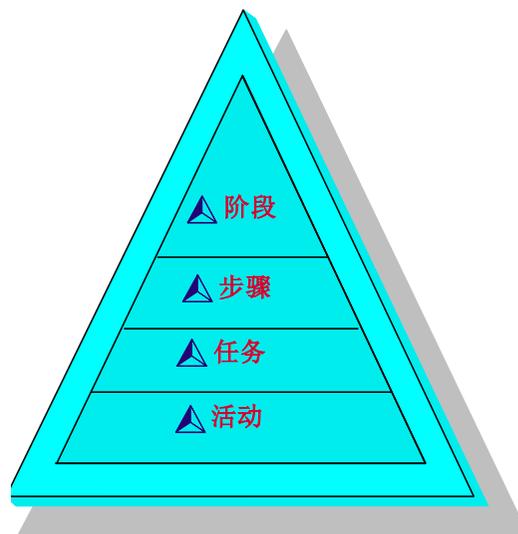
层次性:

- 是否根据产品的特点进行了分层;
- 每项活动是否分解到个人、时间不超过一周;
- 各层次之间配合关系是否明确;
- 特性是否归类

合理性:

- 计划进度是否符合市场需求;
- 技术难度及解决情况是否支撑;
- 资源需求是否合理;
- 资源需求是否可以保证;
- 各阶段、步骤、任务的时间安排是否合理;
- 关键物料的货期是否影响计划;
- 是否符合流程;
- 是否设置了关键路径和里程碑;
- 每个活动是否有结束的标志。

进度、质量、成本、
约束、资源、文档、
配合、里程碑



- ✓项目计划形成之前，最好先画WBS表（Work Breakdown Structure）
- ✓WBS主要原理是：将任务逐级分解直至个人，在矩阵中体现为：先确定横向有多少结点，再将每一结点任务逐渐细化直到个人
- ✓工作分解图（WBS）实际上就是将一个复杂的开发系统分层逐步细化为一个个工作任务单元，这样可以使我们将复杂、庞大的、不知如何下手的大系统划分成了一个独立的我们能预测、计划和控制的单元，从而也就达到了对整个系统进行控制的目的。

关于WBS

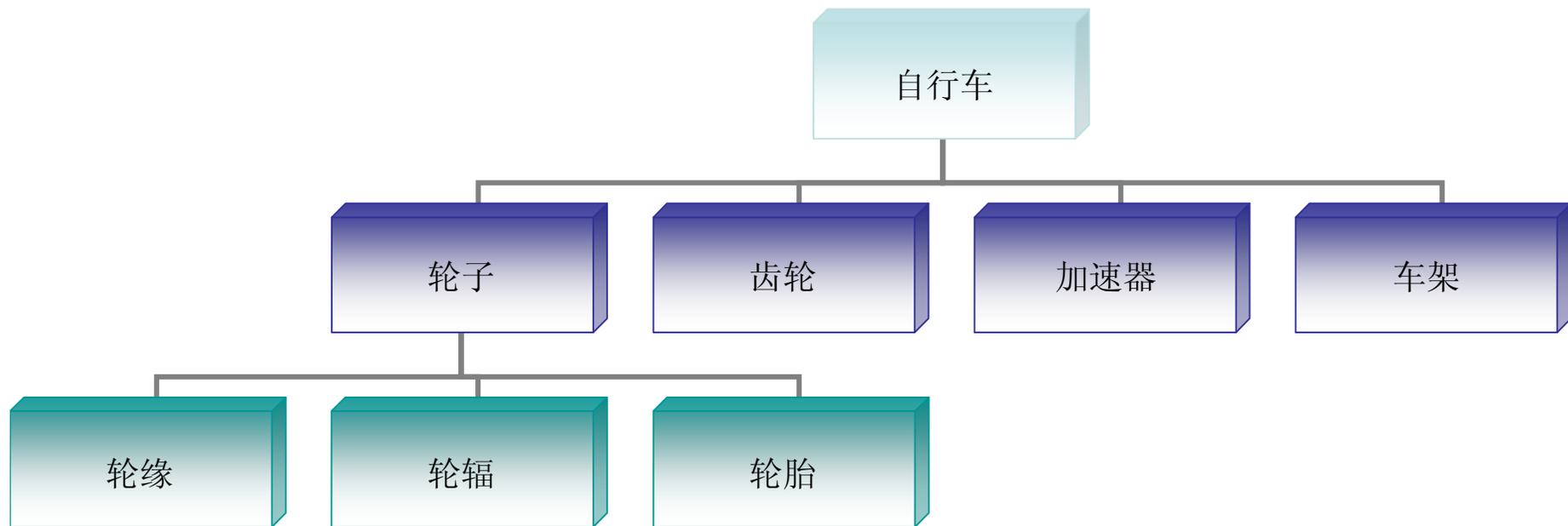
WBS作用

- 责任明确，到员工/团队/分包商，易于管理及追踪
- 可交付成果可以被分解成子交付成果
- 最低层次的子交付成果 被称为工作包 (Work Package) 80hours 原则
- 清晰的里程碑(Milestone)
- 大型项目可以在第一层按项目生命周期分解，以下则按可交付成果分解
- 被用于确定所需资源，确定责任归属，帮助定义项目工作顺序及估计项目时间
- 由上至下或由下至上

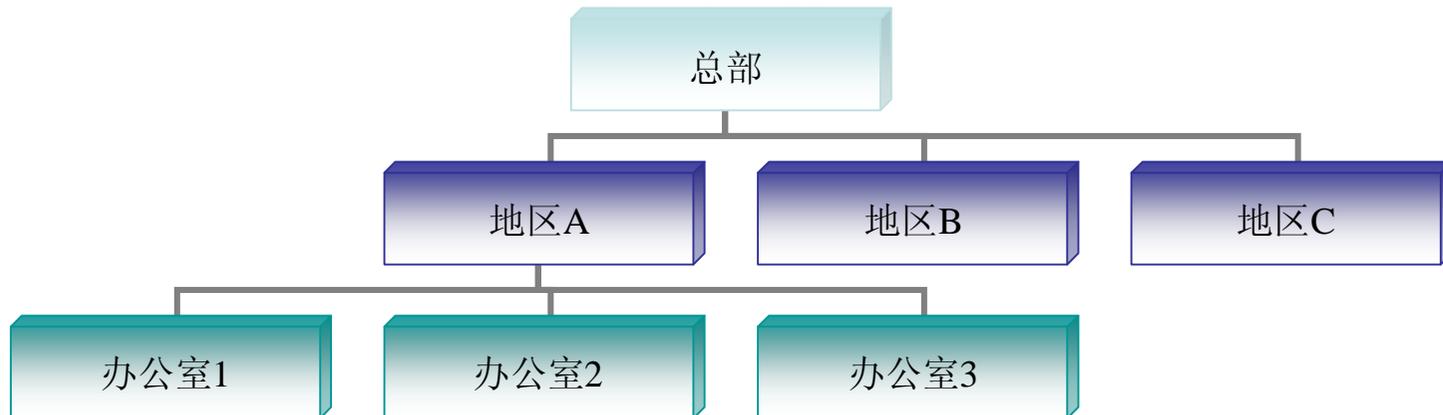
分解过粗？有什么问题？ 分解过细？有什么问题？

- 能够估计每个工作包所需的时间、资源和成本？
- 能够发现各个工作、任务之间的相关关系？
- 能交给具体人去负责？

WBS的类型



按组成部分



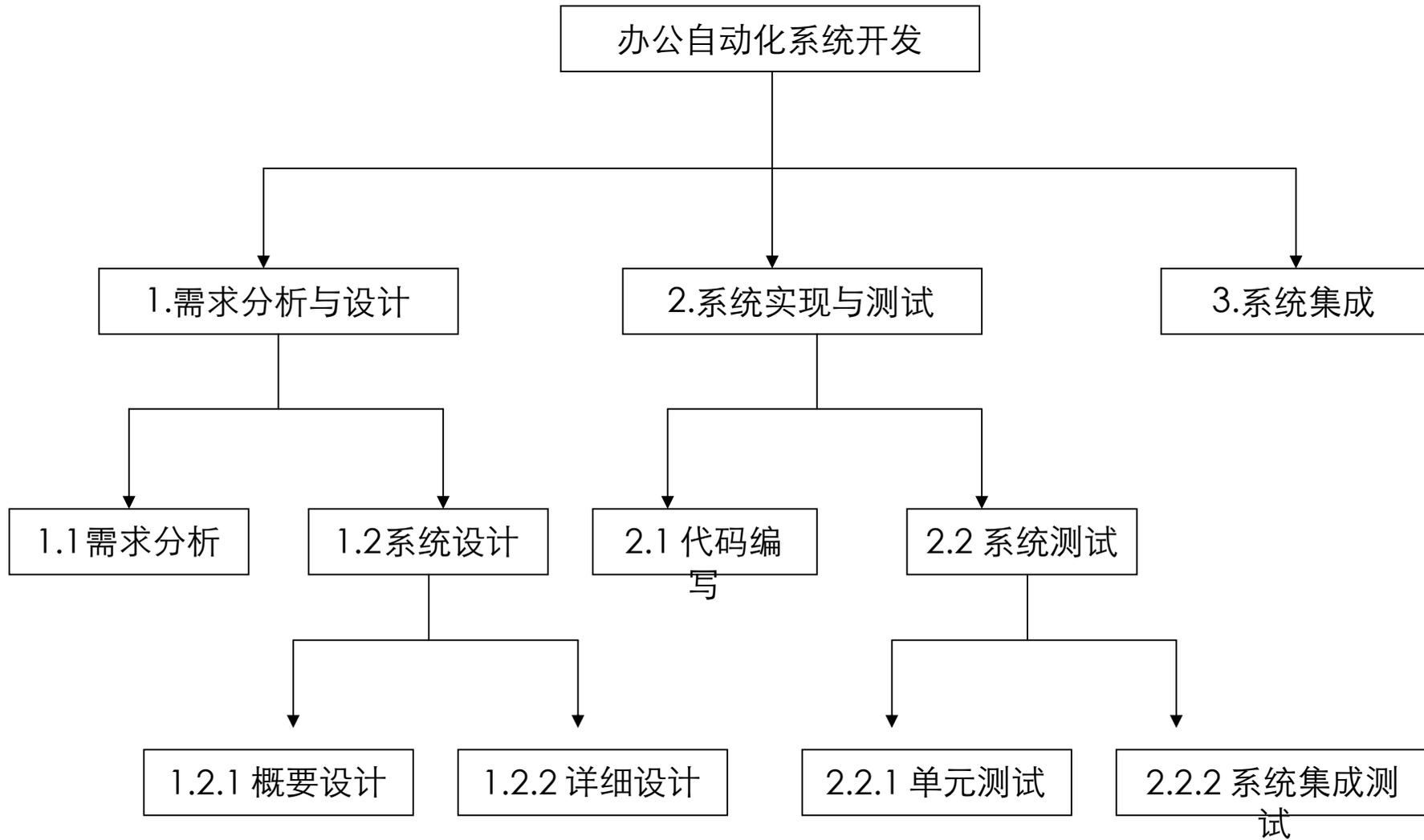
按地理区域

P76

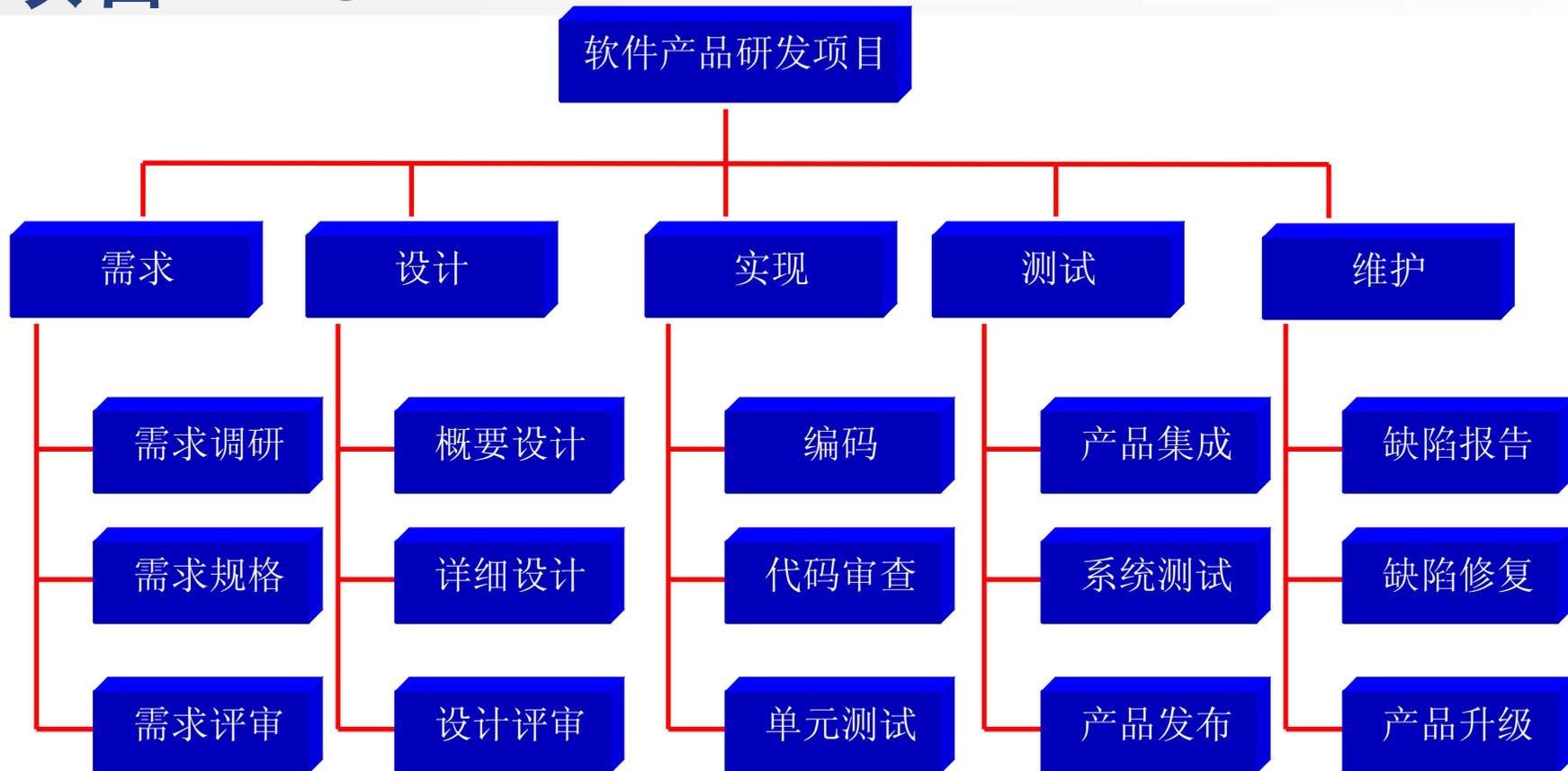
WBS的表示方法 — 列表式

办公自动化系统开发项目的WBS	
1	1. 需求分析与设计
2	1.1 需求分析
3	1.2 系统设计
4	1.2.1 概要设计
5	1.2.2 详细设计
6	2. 系统实现与测试
7	2.1 代码编写
8	2.2 系统测试
9	2.2.1 单元测试
10	2.2.2 系统集成测试
11	3. 系统集成

WBS的表示方法 — 树状结构



项目WBS



项目模拟（课堂实战） WBS分解

- 各小组根据选择的项目，编写该项目的：

1、项目目标书(POS)

2、三级WBS分解

时间：30分钟

- 上台陈述

- 可选案例：

- 1、学员工作中的具体项目
- 2、世界屋脊---西藏行
- 3、我爱我家(家装)
- 4、花好月圆(广东)
- 5、初为人父母

评分： 10

内容 20%

形式 20%

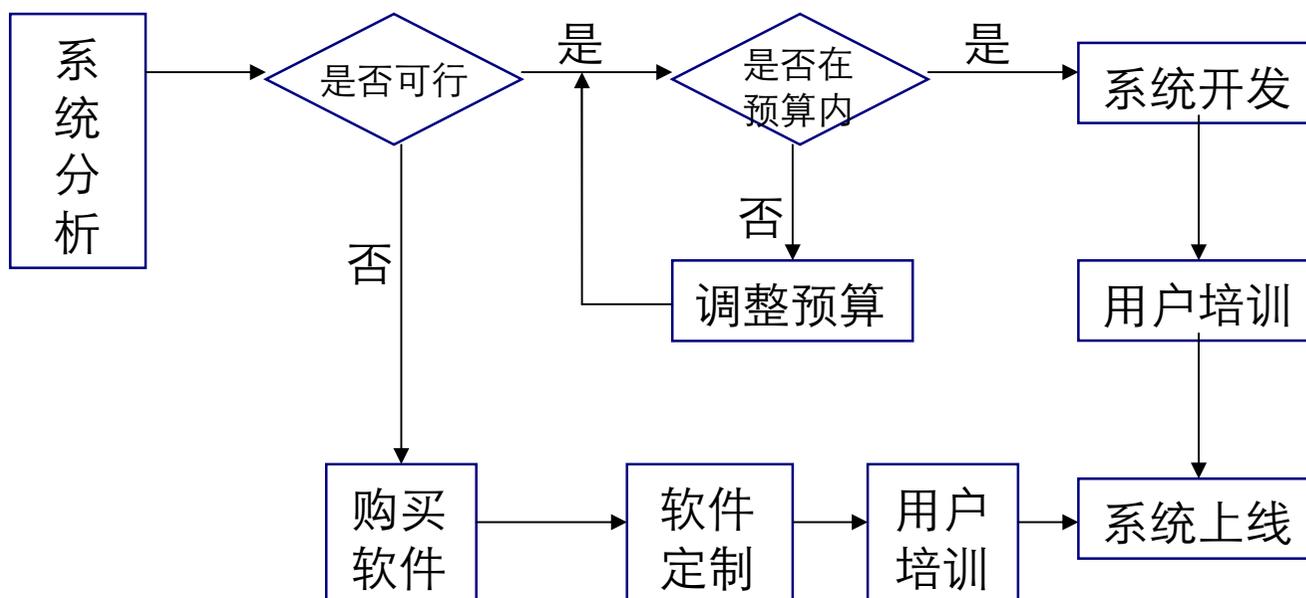
创新 20%

团队 20%

演讲人 20%

图形评审技术

GERT 图例：采用类似流程图的方式来描述项目中的分支活动或回路活动



课堂练习：制作箭线式网络图

活动	紧前活动	活动时间(周)
A	-	2
B	A	6
C	B	3
D	-	4
E	B	5
F	DE	4

课堂练习：粉刷房间的项目

要求：有三个房间要求油漆，其中包括

- 准备房间(打扫)
- 粉刷
- 油漆
- 三者之间是FS的关系

条件

- 有三个熟练工：一个准备，一个粉刷，一个油漆

若每个工人做上述一项工作的时间是一天，则完成三个房间油漆的最长和最短时间分别是多少？

计划历时估算工具与技术

专家类比估算法 (葛洲坝-三峡)

- 也称为“自上而下的估算”，是指以从前类似活动的实际历时为基本依据估算未来活动的历时
- 经常在项目详细信息有限的情况下使用，像项目早期阶段
- 是专家判断的一种形式
- 类比估算法在以下条件下非常可靠：
 - 和以前活动在本质上而不是表面上相似
 - 估算者掌握必要的专门技术

□ 根据工作量估算

- 由工程/设计所确定的每一特定类型工作所需完成的工作量（例如，图纸数量、电缆米数、钢铁吨数等等），乘以生产率（例如，每张图纸所需小时数、每小时电缆米数等等），所得结果可用于估算活动历时

PERT技术

PERT（网络计划评审技术）是以网络图的形式制定计划，求得计划的最优方案，并据以组织和控制开发进程，达到预定目标的一种科学管理方法。

- 1、用网络图来表达一项开发计划中各工作（阶段、模块等）的先后顺序和相互关系；
- 2、通过计划找出计划中关键工序和关键路线；
- 3、通过不断改善网络计划，选择最优方案并付诸实施；
- 4、在计划执行的过程中进行有效的控制和监督，保证合理地使用人、财、物，按预定目标完成任务。

“向关键工作要时间，向非关键工作要资源”

计划历时估算工具与技术--PERT法

- 计划评审技术 (Program Evaluation and Review Technique)
- 一种概率方法

期望值 = (最乐观时间 + 4x最有可能时间 + 最悲观时间) / 6

标准差 (σ) = (最悲观时间 - 最乐观时间) / 6

PERT举例

- 单项活动的工期估算
 - 需要给出每项活动的三个估计时间：乐观时间 t_o 、最可能时间 t_m 、悲观时间 t_p 。这种期望工期可以用下面的公式计算：

$$t_e = \frac{t_o + 4(t_m) + t_p}{6}$$

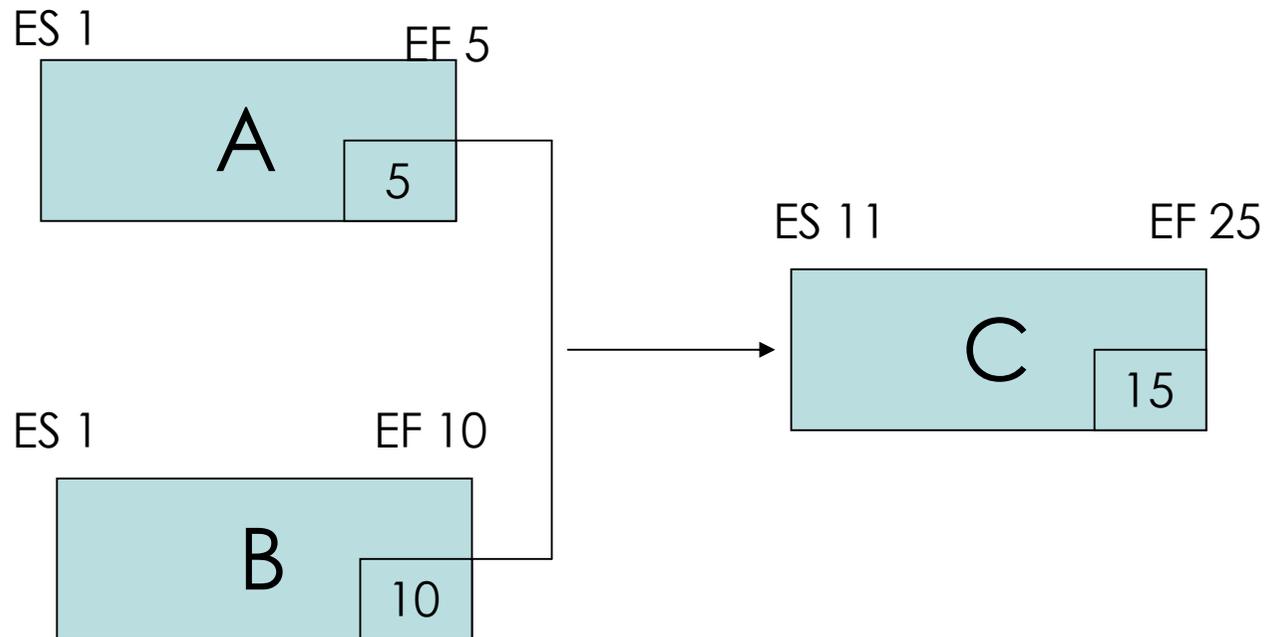
- 总期望工期的计算方法

活动	乐观时间- t_o	最可能时间 t_m	悲观时间 t_p	期望工期 t_e
A	5	4	6	4
B	5	13	15	2
C	13	18	35	20
项目整体	20	35	56	36

项目活动工期估算汇总表 单位：天

前推法计算

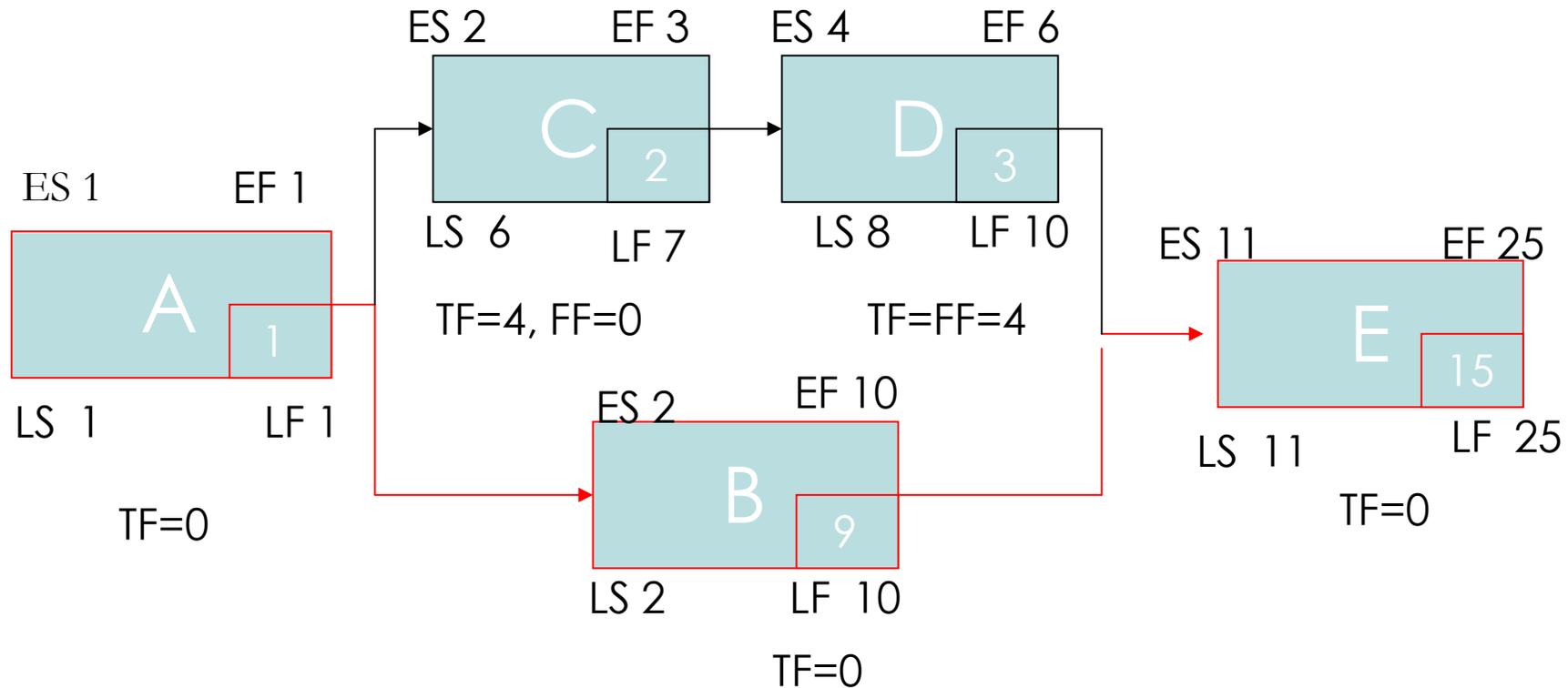
前推法得出活动的ES、EF及项目最早完工时间



最早完成 EF= 最早开始 (ES) +工期- 1

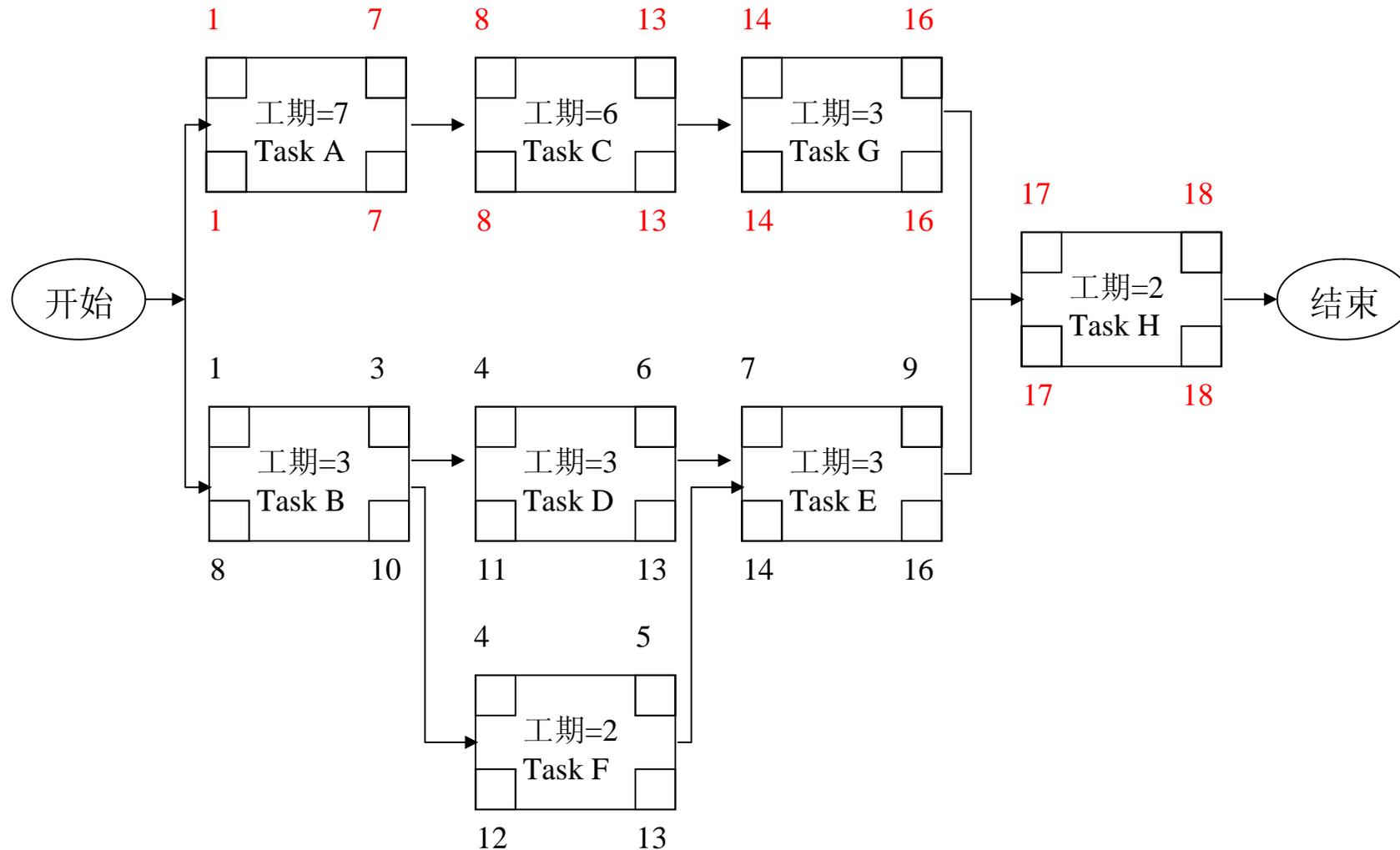
逆推法计算

逆推法确定LS、LF及项目的最晚完工时间



最晚开始 (LS) = 最晚完成 (LF) - 工期 + 1

课堂练习1：关键路径答案



课堂练习：资源平衡

根据以上条件，绘制项目甘特图

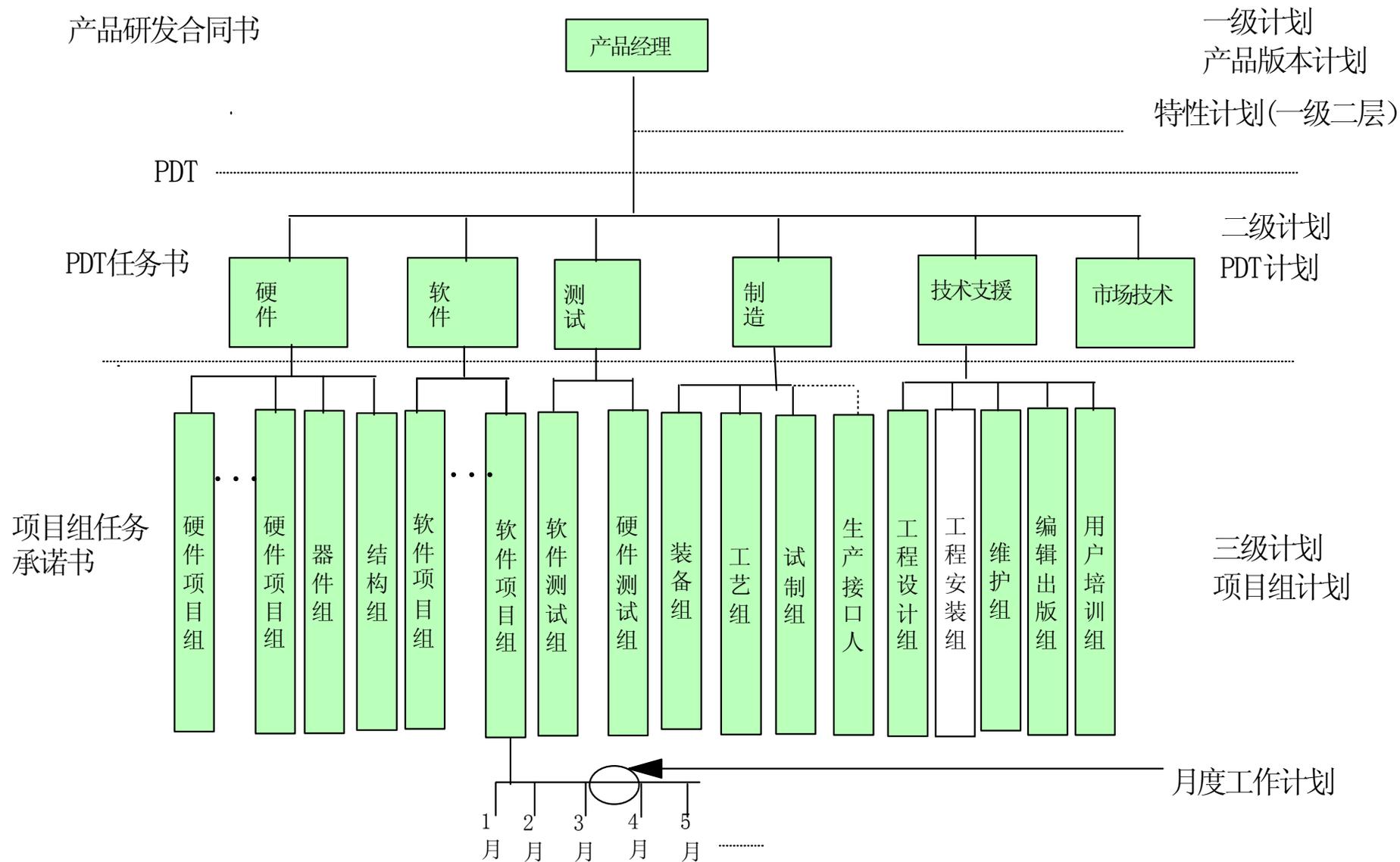
绘制资源柱状图

在没有资源限制的情况下，请进行资源平衡

如仅有两名油漆工人，请进行资源平衡

	活动	时间	油漆工人	前导活动
A	粉刷卧室	6	1	——
B	粉刷地下室	4	1	——
C	粉刷一楼的客厅和其它房间	8	2	——
D	粉刷楼梯和二楼客厅	4	1	C
E	粉刷浴室	2	1	C

产品研发任务、角色与三级计划体系



研发一级计划制定

- 根据产品规格和总体方案划分特性；
- 根据市场的要求或产品战略或其他因素确定版本和特性的计划总目标；
- *产品经理（版本经理）各PDT经理（软件经理、硬件经理、测试经理、制造经理、市场技术经理、技术支援经理等）、特性经理沟通协商，大致确定版本和特性的阶段时间分布。*
- 根据二、三级计划的制定情况，来修正版本和特性一级计划

研发二级计划制定

- 各PDT经理根据产品的一级计划获取二级计划总目标；
- PDT经理与相关的职能部门经理沟通与协商，基本确定各项目组所需的工时和资源、联调测试或试产时间等。
- 根据三级计划的详细制定情况修正二级计划。
- 二级计划既要考虑一级计划的约束，又要考虑三级计划的支持，如果二级计划制定出来后与一级计划有冲突，需要向上反馈，与产品经理（版本经理）协商确定是否要修改一级计划。

研发三级计划制定

- **项目经理**根据二级计划确定项目组的工作计划目标；
- 与项目组骨干成员讨论制定详细的工作计划；
- 制定三级计划同样要考虑二级计划的约束和资源的配合，三级计划是产品计划与资源计划的真正结合点。
- 如果三级计划制定出来后与二级计划有冲突，需要向上反馈，与PDT经理和资源经理协商确定是否要修改二级计划。

案例

PDT软件经理



根据一级计划产品经理让PDT软件经理开发某软件WISDOM，要求制定二级计划，初步最迟要求在1999/11/30开发完成，现在已经是1999/06/25，赶紧找相关人员讨论一下制定进程计划吧！

经过与相关人员讨论，WISDOM软件总共可分成三个模块，A、B、S；得赶紧找资源经理商量并确定项目经理及其他开发人员



资源经理

资源经理

资源经理

PDT软件经理

A项目经理

B项目经理

S项目经理

通过与资源经理沟通，根据工程师的任职资格、过去的开发经验以及技术的复杂程度，进一步分析得出：

- 1、A、B、S三个项目可以并行开发；
- 2、各项目的人员计划为：A项目3人、B项目3人、S项目5人；

.....

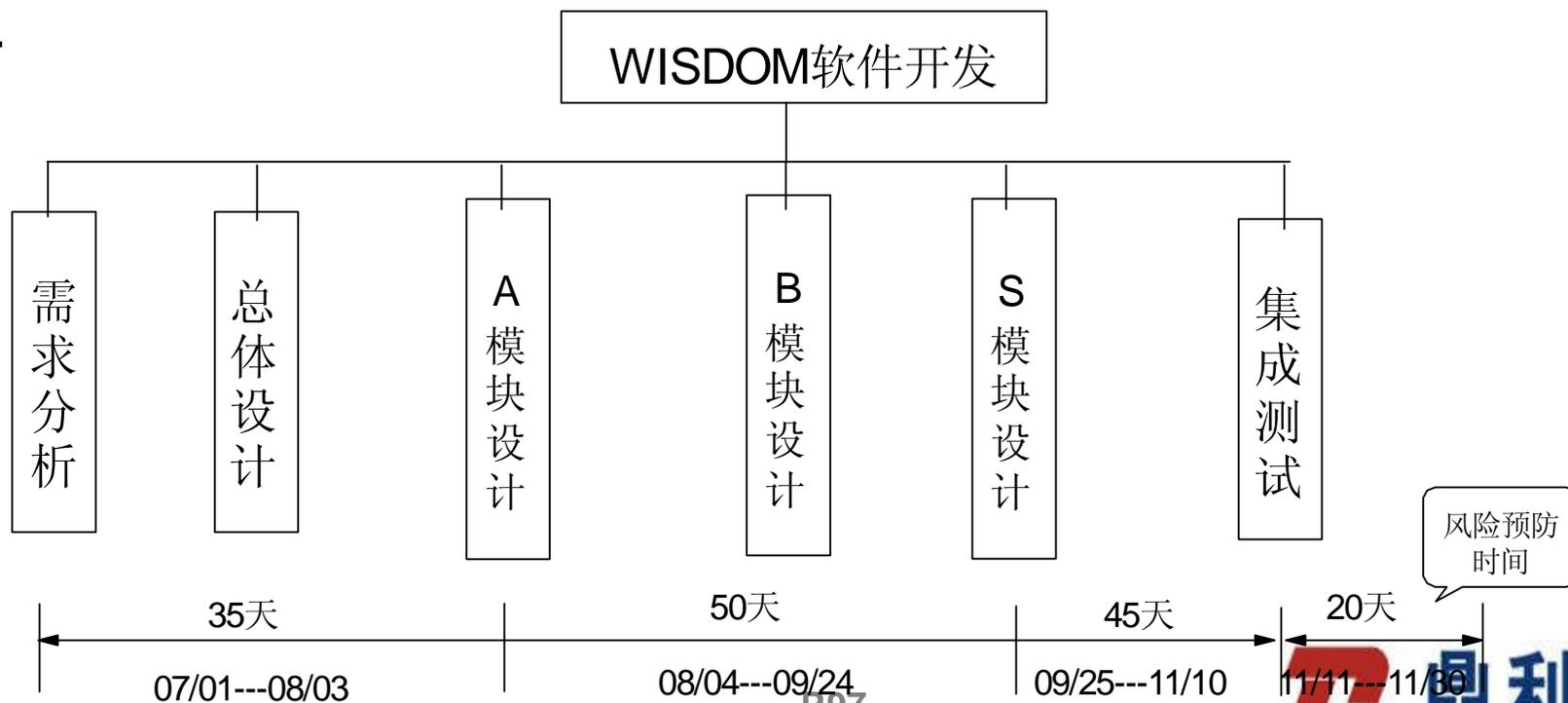


WBS

- 一、五个月的开发时间，预留20天作为抵御风险和突发任务的时间
- 二、对工作任务进行初步的分解；将整个软件模块的开发按照系统分析和设计、实现、测试三个步骤来完成，根据以前的统计结果给出各阶段的经验DEADLINE；
- 三、与相关的资源经理沟通讨论，确定详细的任务时间及调整各阶段的完成时间如下图；
- 四、根据以上的分析作出WBS、PERT图，并调整关键路径；

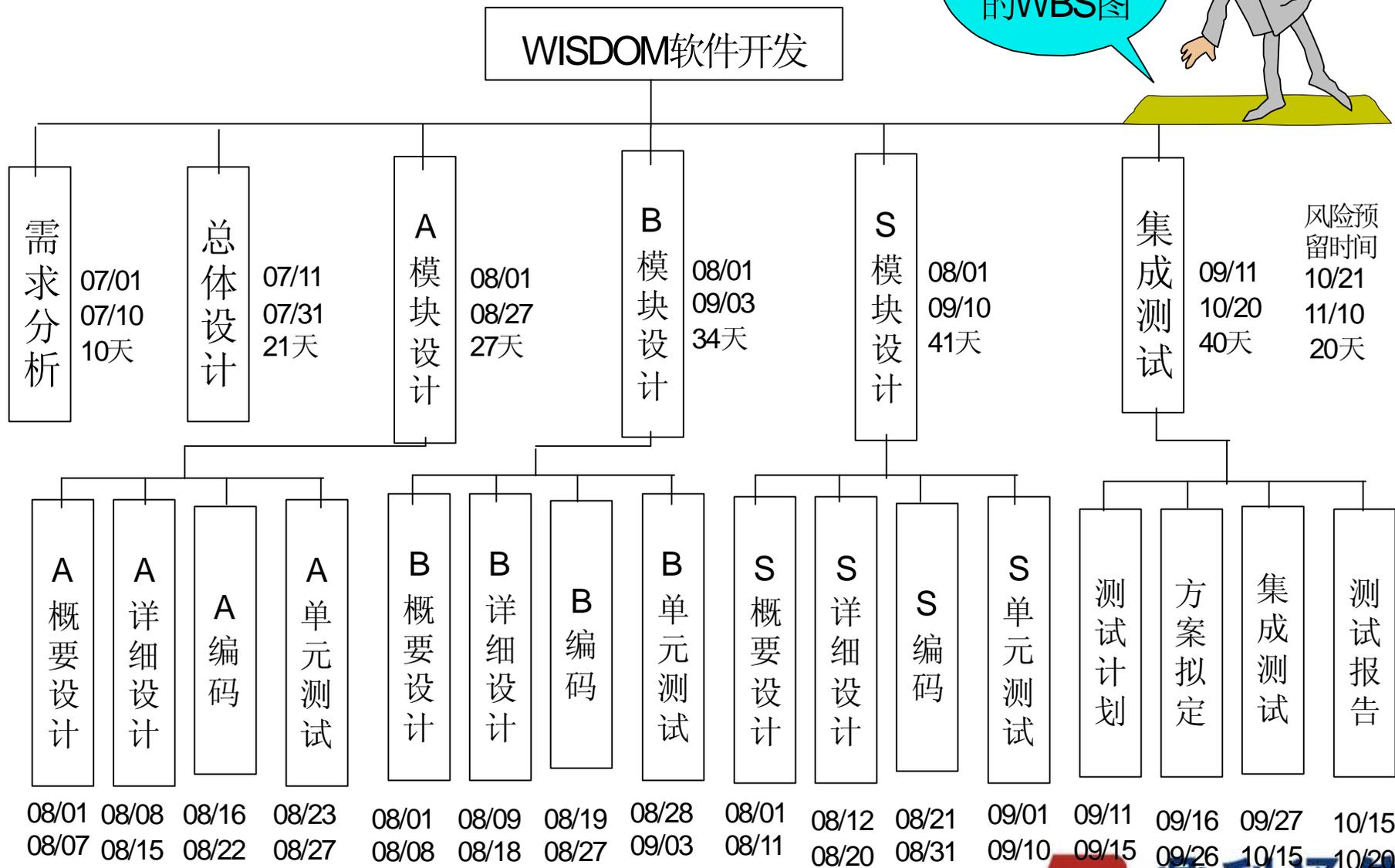


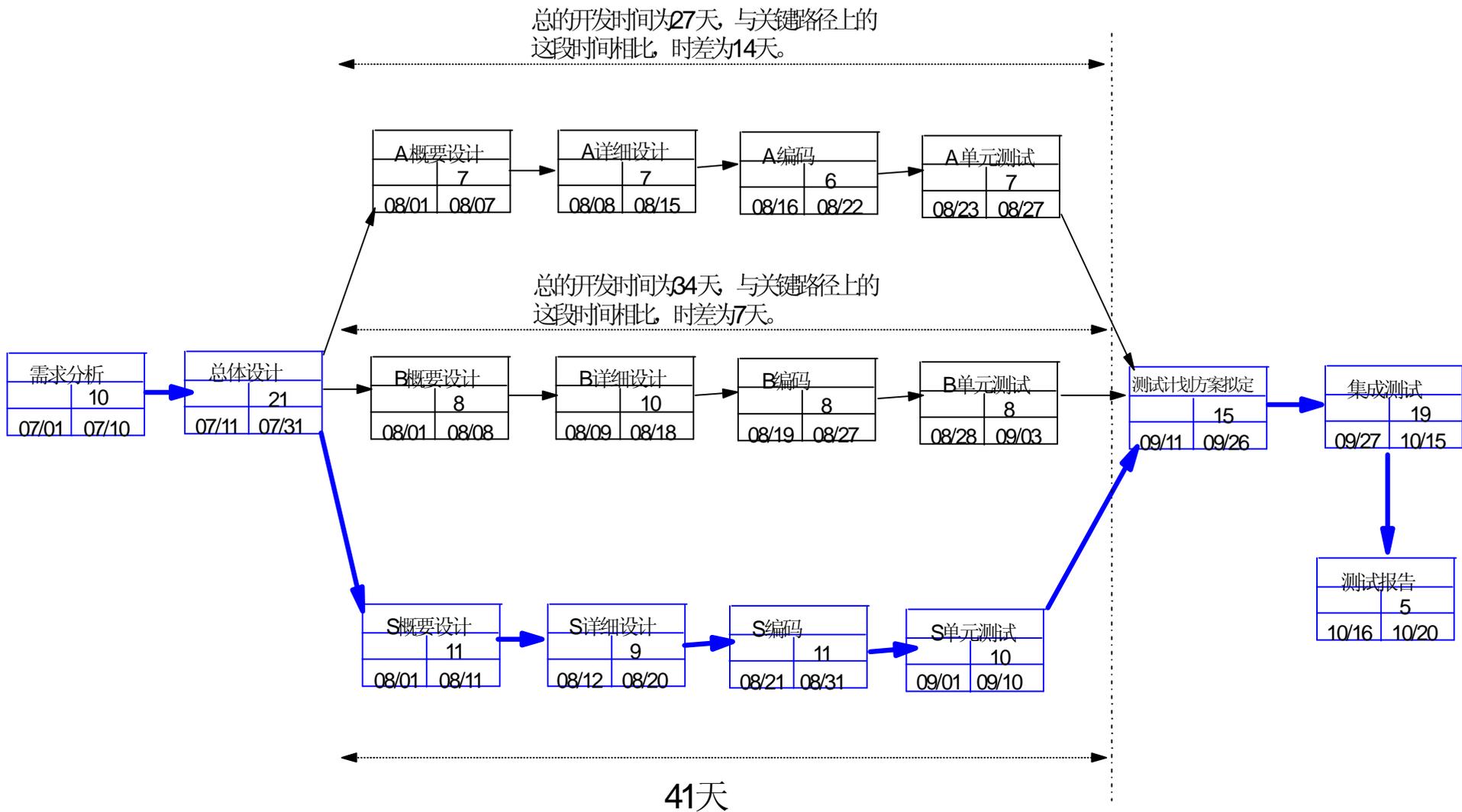
.....



WBS

详细讨论后的
WBS图





经过讨论发现我们在11月10日就可以完成开发!



找产品经理根据二级计划修改一级计划去!



PDT软件经理的二级计划可以提前20天完成，我们一起商量修改一下一级计划.....

软件经理

产品经理

硬件经理



产品计划与资源计划

- 1、产品计划制定出来的同时，必然产生与产品计划匹配的资源需求计划，包含以下内容：
 - 人力资源需求计划；
 - 仪器设备需求计划；
 - 物料需求计划；
 - 环境、场地需求计划；
 - 资源需求计划是产品研发预算的基础，也是进行财经分析的依据。
- 2、产品计划以任务驱动、以目标为导向，但是要受资源约束。
- 3、产品计划是资源线的计划来源之一。资源线的计划来自三个方面：
 - 各个产品线的计划；
 - 培训计划；
 - 业务研究计划；

资源实施对开发进度影响

1、缩短开发任务的方法

要达到缩短开发周期的任务，通常有三种方式：

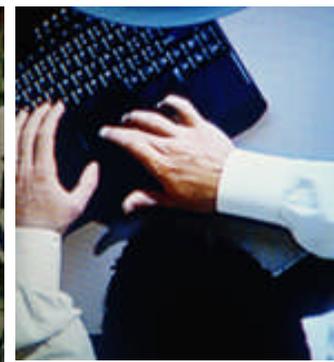
- (1) 多投入资源（人力资源、环境等），特别是对于关键路径上的任务；
- (2) 根据任务的重要程度，取消/延迟某些开发任务；
- (3) 对工作任务进行细分，内部挖潜；

2、向关键路径要进度，向非关键路径要资源

通过向关键路径投入资源等方式，缩短开发进度；在发生突发任务等情况时，我们可以从非关键路径上抽调人员。需要但需要明确的是，非关键路径在一定条件下可转化为关键路径，特别是浮动期短的非关键路径



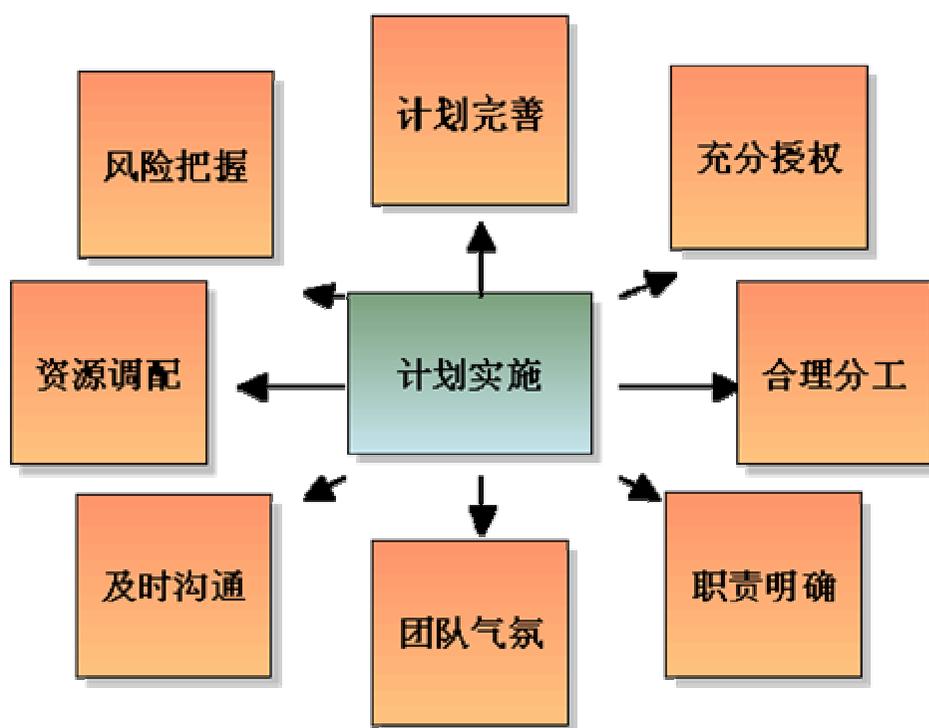
DINGLI COMMUNICATIONS CORP.,LTD.



目录 *Content*

- 一、研发项目与项目管理的基本概念
- 二、研发管理、IPD、CMMI、敏捷的定义及关系
- 三、软件开发模型与工作方法
- 四、研发项目管理及组织
- 五、研发项目计划制定
- 六、研发项目计划控制**
- 七、在KD研发中心的某一天
- 八、研发项目精细化管理
- 九、KD研发中心项目管理体系

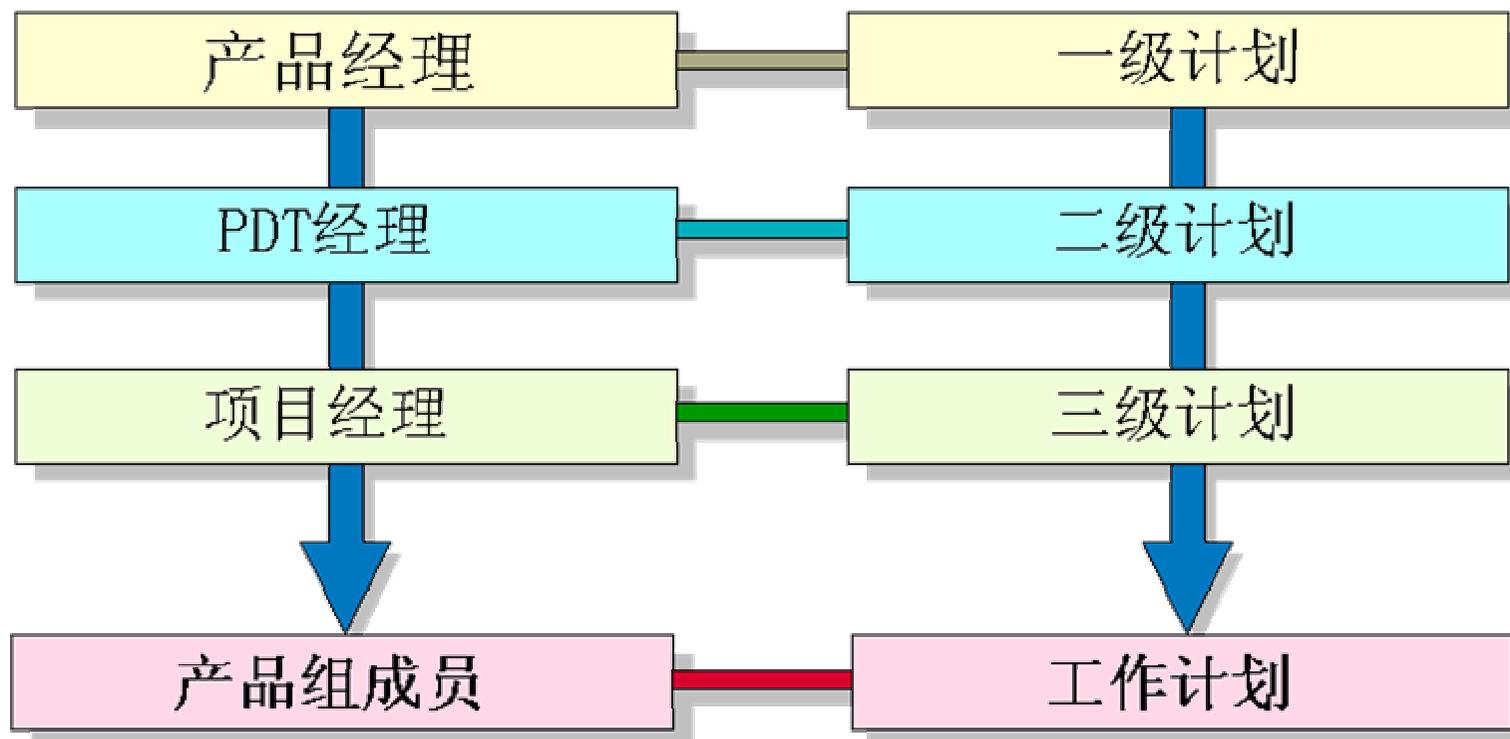
一张完美的图纸 ≠ 一栋坚实的大楼



计划是会变化的，变化来源于：

- 市场需求变化；
- 进度异常；
- 预估不准确；
- 配合问题；
- 人员变化；
- 资源不到位；
- 项目位置变动；
-

分层实施、分层监控



- 各级监控点的设立遵循两个原则：
 - A、重要的里程碑
 - B、时间间隔比较合理
- 监控计划的表现形式为：计划监控总揽图和计划监控一览表。
- 计划监控总揽图将各级计划的关键点浓缩在一起，直观，便于控制。同时，各监控点在时间上也形成对应。
- 通过计划监控一览表，严格定义了每一监控点的完成标志，以使监控点不会产生歧义性的理解。

合同书和任务书明确了各方的责任和承诺，是监控、保证产品各级计划执行的主要形式。



合同书/任务书

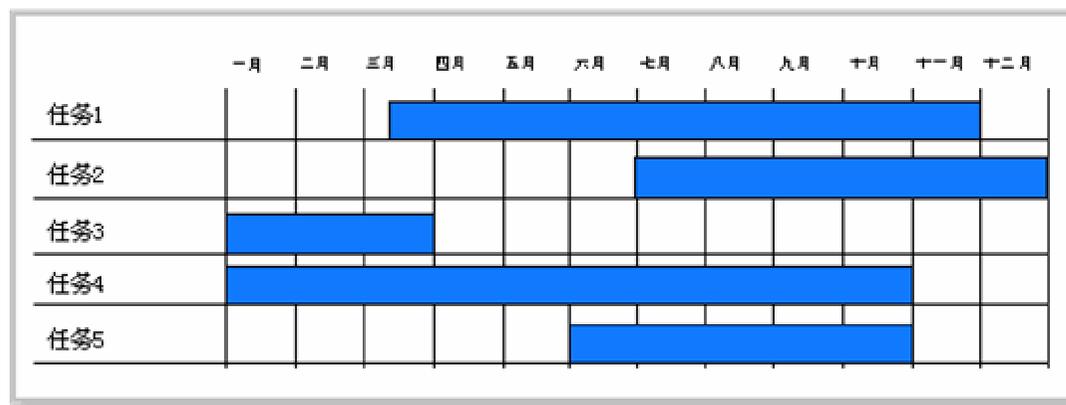
- 合同书/任务书主要规范合同双方的责任和权利。对合同的执行者，主要承诺如下目标：
 - 进度目标:指开发的进度方面，主要考核指标为计划的完成率、物料及时齐套率等
 - 质量目标：主要指稳定性方面，其考核指标包括故障率、单板直通率、各种问题反馈、处理率、文档合格率等、需求规格重大修改率等
 - 成本目标：主要指设计成本降低方面，其考核指标为设计成本降低额、呆死料发生额等
 - 人均毛利额：主要指人均销售毛利额
 - 器件复用提高、独家供应商减少方面的目标
- 对合同的发包者，其承诺的主要目标如下：
 - 资源及时提供，特别是人力资源的及时提供
 - 保证及时组织评审
 - 保证及时提供相关文档资料
 - 保证及时处理跨部门问题等

计划更改与状态转移

- 计划更改须经过评审，其评审批准部门同计划制定。一级计划更改须填写一级计划更改单，并修订相关计划。
- 原则上，一级计划不予修订，二、三级计划要及时修订滚动。以保证一级计划最终按目标实现。
- 在版本立项通过后，即为该版本建立状态转移表，直至版本转产，状态转移表是一级监控的检查档案。

预警系统

您好，现在向您播报消息：今天是8月1号，按计划您准备于本月20进行的评审应该从今天开始准备，请您预先作好工作安排！



• 正规控制

在每周末、每月末或每个阶段末进行情况汇报和检查等，通过PERT图和GANTT图以及预算报告和工作总结及阶段评审的报告等及时发现问题和进行评审

• 非正规控制

通过工作之外的交流和沟通进行控制，在非正规控制的场所要比在办公室更坦率、更诚实，这样能了解到正在酝酿的问题，这要比等到这些问题出现在情况报告中或某次会议上快得多

项目控制的五个步骤

- 1、及时掌握最新情况和项目进展
- 2、分析计划进度和质量产生偏差的原因
- 3、处理偏差
- 4、公布修改方案及滚动的计划
- 5、周知管理部门

项目经理如何召集会议

为明确计划和方案，项目经理应该定期不定期召集例会，例会可以讨论以下问题：

- 里程碑计划为什么没有完成？
- 其影响如何？
- 工作何时可以完成？
- 是否需要替补行动计划？
- 何日才能回到计划进度上来？

项目情况回顾检查会

回顾检查会准备：

- 明确界定会议的目的
- 议程要明确，并要提前进行准备
- 邀请必要的人到会，尤其是关键人物入会
- 如果有重要人物参加，最好让你的发言人预先演习
- 以对你最有利的方式组织项目回顾检查会
- 遵守议程
- 自上次回顾会以来的主要成就
- 进度状况、成本状况（实际与计划相对照）
- 重大问题及行动计划
- 下个阶段的计划
- 特殊议题（具有紧迫性的议题）
- 总结由本次会议产生的各行动事项，明确责任人和时间
- 切勿超时

管理变更（续）

管理变更的流程

1. 要认识到变更是不可避免的，并做好相应的安排
2. 创建需求底线
3. 建立一个专门的渠道管理变更
4. 通过一个变更控制系统获取变化信息
5. 分级管理变化

管理变更（续）

步骤 1 — 认识到变更是不可避免的，并做好相应的安排

- 作出管理变更的计划
- 正规化变更请求
 - 来自用户
 - 来自设计人员
 - 来自开发人员

步骤 2 — 创建需求底线

- 对所有已知的系统需求创建需求底线
- 对相关产品进行版本控制
- 发现变更比例达到1%-4%是很正常的

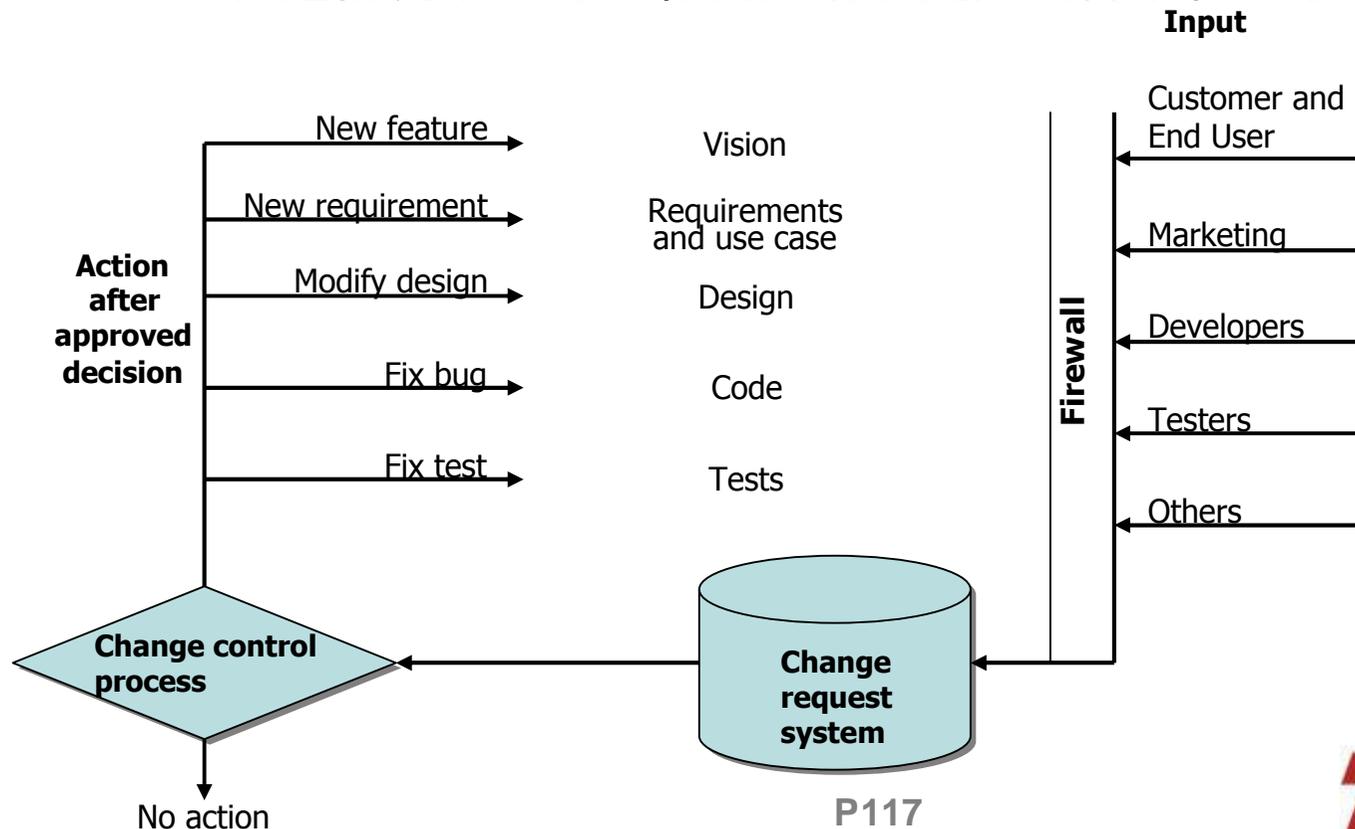
步骤 3 — 建立一个专门的渠道管理变更

- 从客户的角度，他们希望这些变更不被看作是正式修改进度和预算
- 每一个变更都通过专门的通道进行管理，从而判断该变更的影响并作出正式的决定，这一点是非常重要的。
- 正式通道
 - 项目倡导者或者项目经理（小型项目）
 - 变更控制委员会（CCB）的成员，他们是权威并具备技术能力（大型项目）

管理变更（续）

步骤4 — 采用变化控制系统获取变更信息

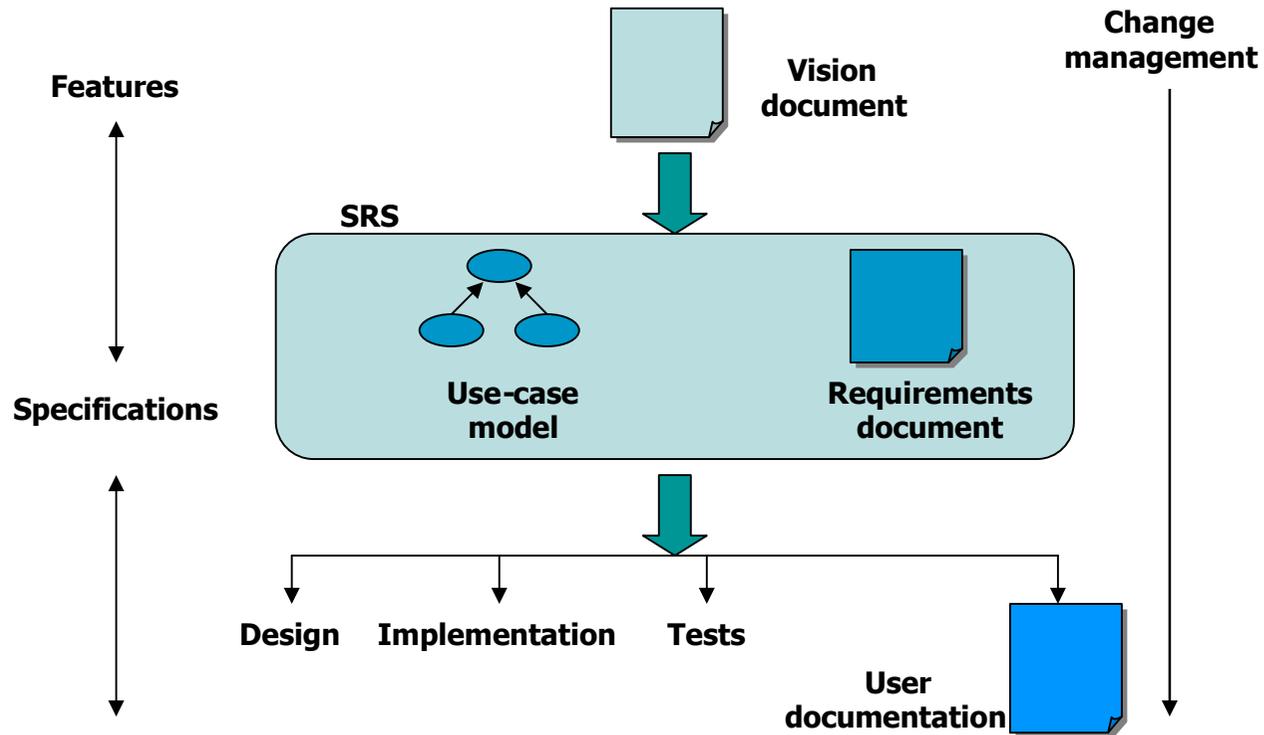
- 实现一个正式的方法，获得所有系统变更请求
 - 集中式数据库管理变更请求
 - 状态跟踪
 - 自动通知受影响的团队
 - 合适情况下，可以将变更请求提交到需求管理系统



管理变更 (续)

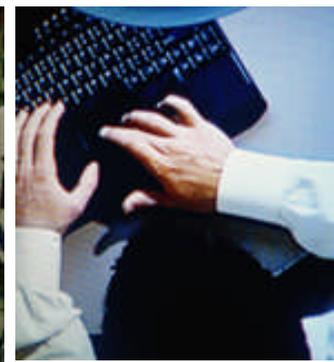
步骤 5 — 分层管理变更

- 一个程序员是没有权力代表用户在代码中直接加入新的功能和需求的





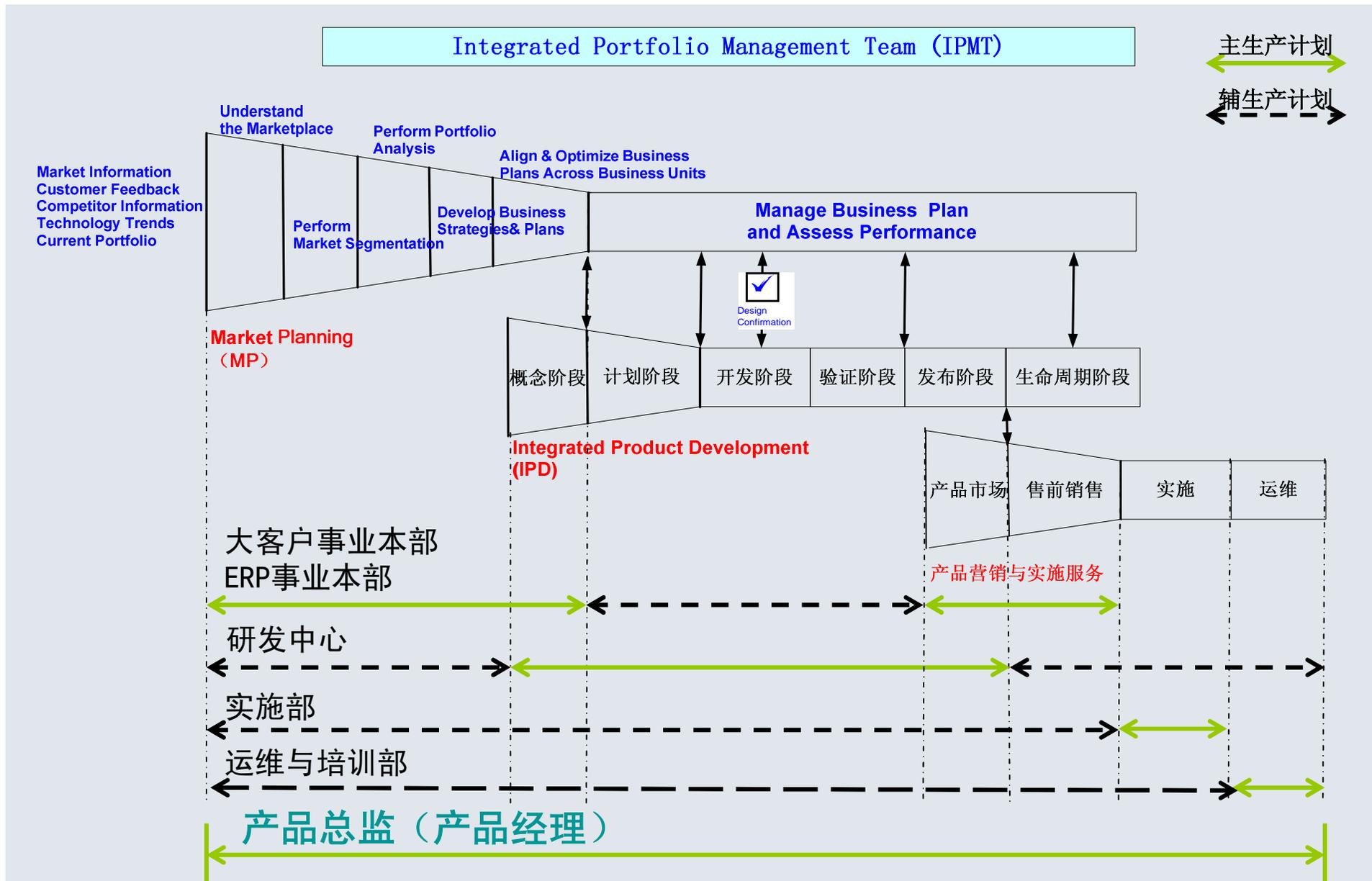
DINGLI COMMUNICATIONS CORP.,LTD.



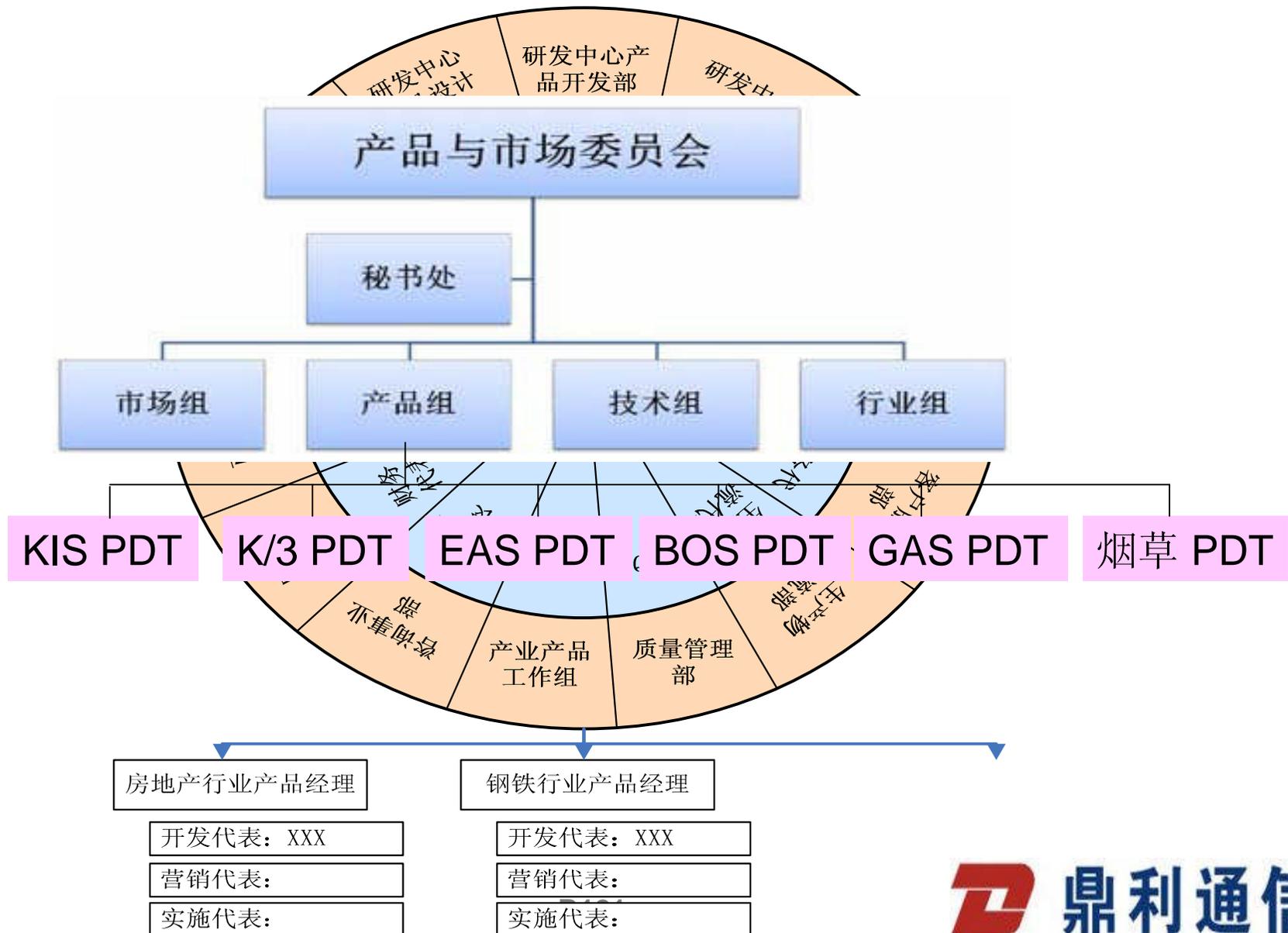
目录 *Content*

- 一、研发项目与项目管理的基本概念
- 二、研发管理、IPD、CMMI、敏捷的定义及关系
- 三、软件开发模型与工作方法
- 四、研发项目管理及组织
- 五、研发项目计划制定
- 六、研发项目计划控制
- 七、在KD研发中心的某一天
- 八、研发项目精细化管理
- 九、KD研发中心项目管理体系

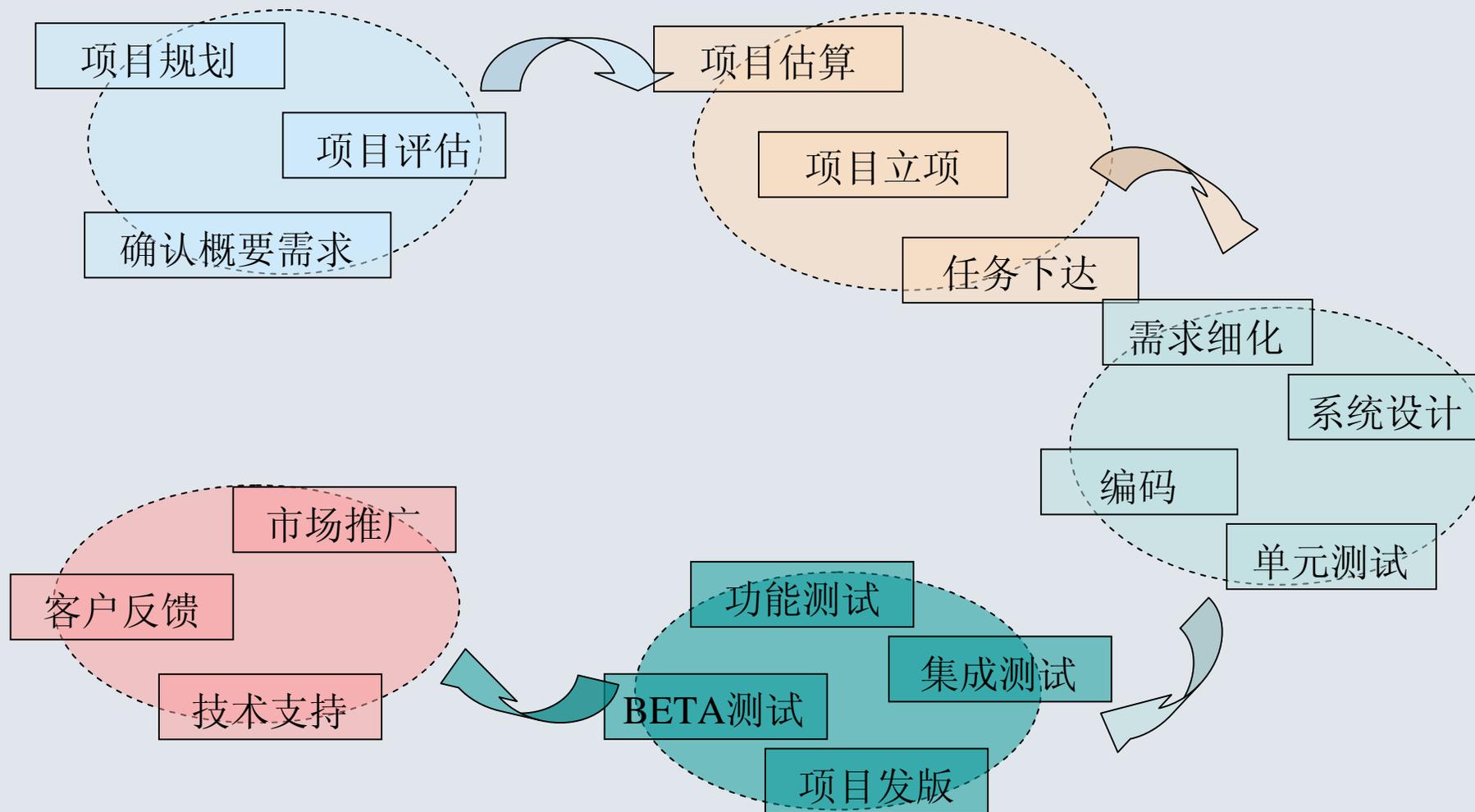
集成研发流程-IPD



KDIPD组织模式

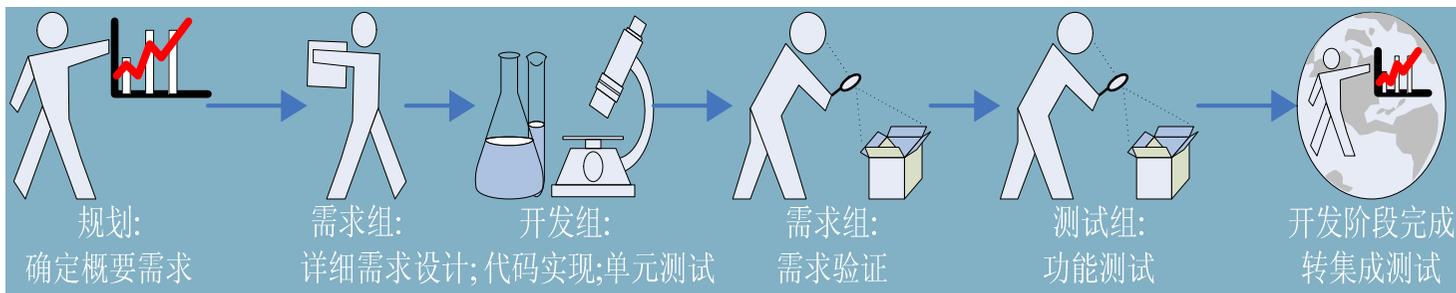


产品研发流程分解



开发迭代阶段

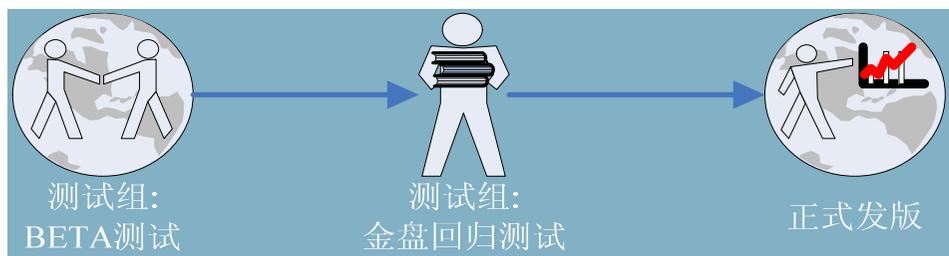
研发阶段



集成测试



BETA测试;
发版



产品研发计划体系

一级计划：产品年度计划

产品年度计划

产品线: EAS 产品总监: 隋同建 日期: 审批人:

序号	计划编号	工作任务 (将策略分解成多项任务)	责任环节	任务负责人(具体到人)	Q1		Q2	
					M1	M2	M3	M4
1		收款/出货套数/新增客户数						
1.1	FA001	EAS6.0产品研发(提供生产成本、协同、流程制造等功能)	产品部	隋同建, 张大亮	开发	发布		
1.1.1	FA002	管控风险 稳见未来——2009' 讨会(包括6.0发布会)	产品市场/产品部	隋同建, 张大亮		△	△	
1.1.2	FA003	售前顾问/销售培训						
1.1.3	FA004	流程制造样板客户的树立—完善						
1.1.4	FA005	生产成本样板客户的树立—完善						
1.1.5	FA006	协同样板客户的树立—大连亿成						
1.1.6	FA007	实施顾问产品培训						
1.2	FA008	专场市场活动						
1.2.1	FA009	发布风控演示包, 举行风险与P						

二级计划：滚动执行计划

滚动执行计划

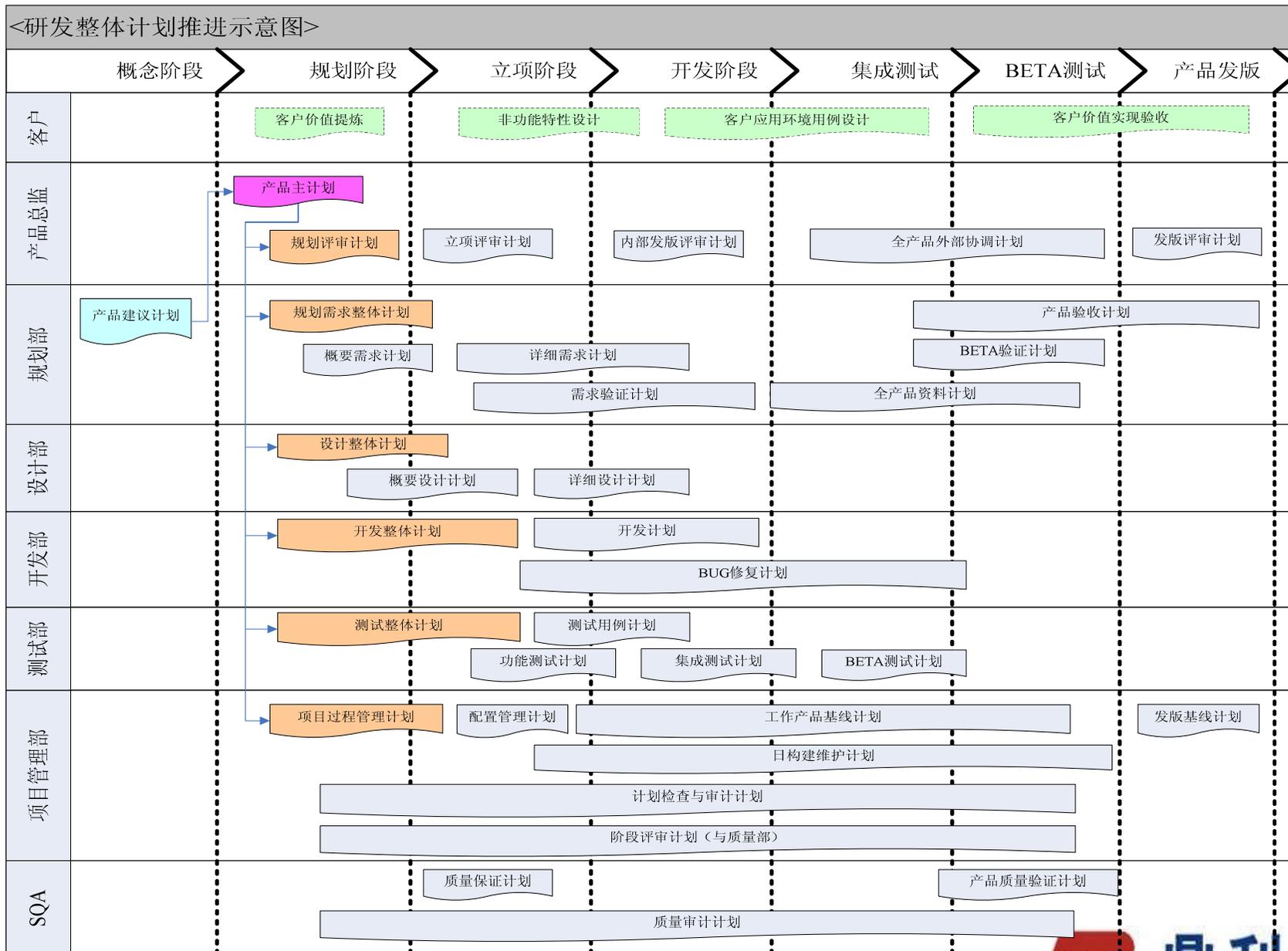
产品线: EAS 产品总监: 隋同建

序号	计划编号	活动内容	开始时间	结束时间	工作任务 KPI	责任人	团队成员
1		EAS产品部					
1.1	FA001	EAS6.0产品研发 提供生产成本、协同、流程制造等功能			收款/出货套数/新增客户数	隋同建, 张大亮	
1.1.1		BETA测试阶段			收款/出货套数/新增客户数	隋同建, 张大亮	
1.1.2		关闭阶段			收款/出货套数/新增客户数	隋同建, 张大亮	
1.2	FA017	EAS6.1产品研发 提供离散制造, 多结算中心、业务预算、风险与内控、委外加工、HR自定义报表单据流程等			收款/出货套数/新增客户数		
1.2.1		概念阶段			收款/出货套数/新增客户数		
1.2.2		规划阶段			收款/出货套数/新增客户数		
1.2.3		开发立项阶段			收款/出货套数/新增客户数		
1.2.4		需求阶段			收款/出货套数/新增客户数		
1.2.5		设计阶段			收款/出货套数/新增客户数		

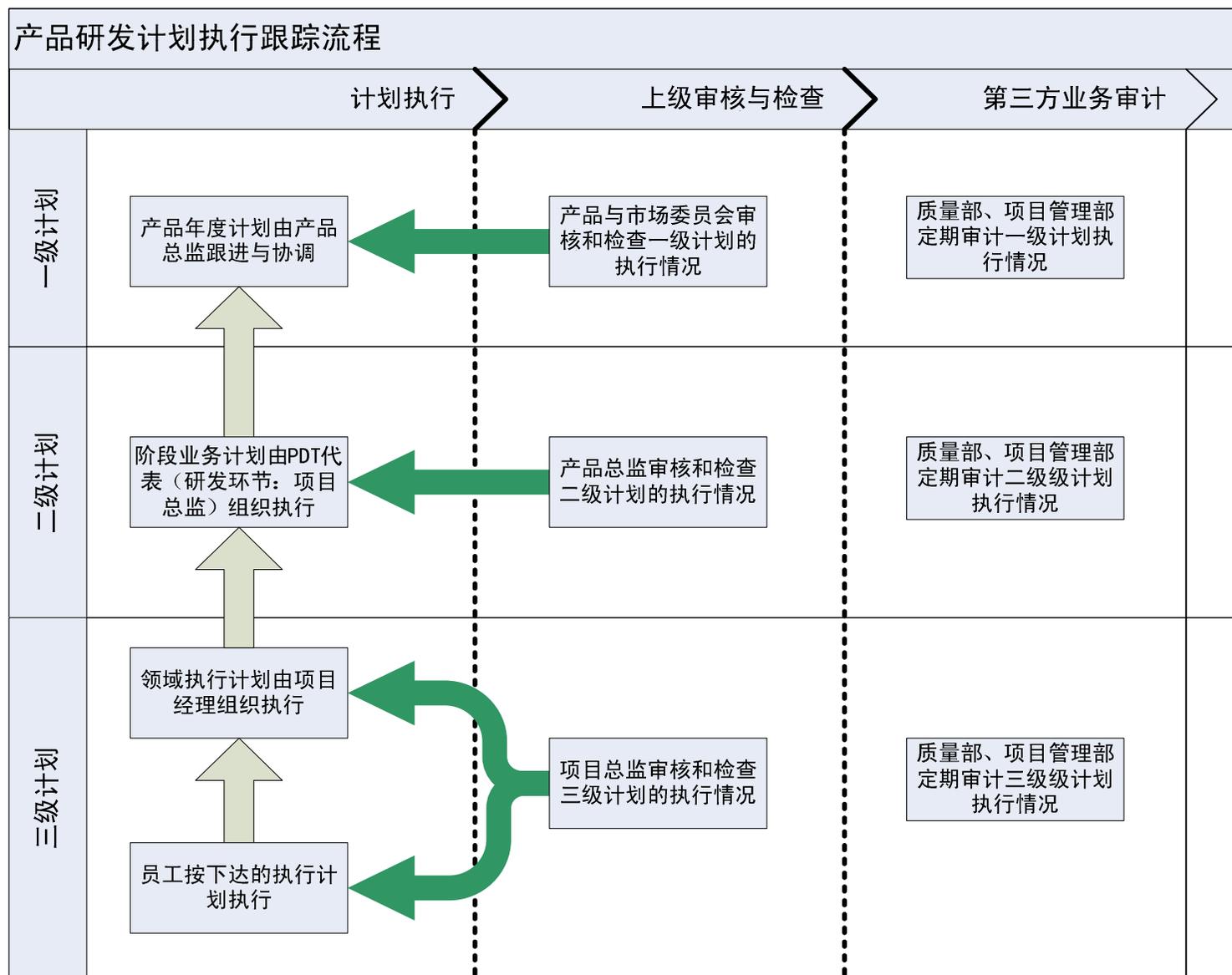
三级计划：研发执行计划

PRJ00168503	ar ap002						
PRJ00169840	ar ap002_1_1	SCM_成长版_取消项目管理控制	完成	开发	应收款管理		通过
PRJ00171018	ar ap002_1_1_1	SCM_成长版_取消项目管理控制	完成	测试	应收款管理		
PRJ00168073	ar ap003	SCM_成长版_取消单折扣	进行中	需求	应收款管理		

项目计划分解与推进



计划执行跟踪



需求(规划)人员的一天

- 1 登录RMP系统, 检查有无本系统的客户反馈提单(答复提单)
- 2 登录DMP系统, 检查和确认昨天需求评审评审委员提出的评审缺陷(需求评审)
- 3 登录DMP系统, 检查有无提交自己验证的开发任务或补丁任务, 从DailyBuild获取最新环境验证, 通过后提交测试
- 4 登录DMP系统, 检查有无提交自己的需求缺陷, 及时组织讨论确认
- 5 给负责的原型客户(实施)电话联系, 了解现场客户应用情况, 约好下次去客户现场调研的时间安排

开发人员的一天(接下来...)

- 1 登录DMP系统, 检查有无指定自己的BUG
- 2 从Dailybuild获取最新环境, 将昨天已经开发完成的功能点, 执行单元测试, 通过后, 通过DMP流程提交需求验证
- 3 登录DMP系统, 检查昨天提交测试的补丁任务是否已经通过, 完成代码基线, 提交补丁发放
- 4 配合项目经理, 对昨天发现的典型BUG及今天修改的内容组织code review. 发现问题及时修正
- 5 将今天修改的程序及时CHECK IN, 并提交编译打包

测试人员的一天

- 1 检查凌晨的自动化测试结果, 发现异常及时通报相关人员处理
- 2 登录DMP系统, 检查提交自己验证BUG, 从Dailybuild 获取最新环境验证, 通过后及时标识通过状态
- 3 登录DMP系统, 检查有无提交自己测试的开发任务, 从Dailybuild 获取最新环境, 根据测试用例进行测试
- 4 登录DMP系统, 检查有无提交自己验证补丁任务, 从补丁服务器获取最新补丁环境验证, 通过后提交开发安排后续发放
- 5 和其它测试人员交流和分析当天的测试状况, 发现异常情况及时通报项目经理和开发人员

项目经理的工作

检查昨天DailyBuild, 自动化测试, 测试趋势分析报告数据,及时跟进发现的问题

根据DMP数据检查和分析目前的项目进度和质量状况, 及时调整工作策略和计划

主持Bug专家会诊等专题讨论

审核并提交项目变更

及时更新项目的计划、日程表、产品规格书、风险控制列表

关注重点功能的开发进度和测试进度

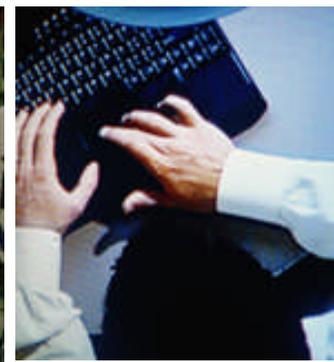
根据阶段目标及时检查项目状态, 定期汇报项目状态

.....

PM is the communication hub!



DINGLI COMMUNICATIONS CORP.,LTD.



目录 *Content*

- 一、研发项目与项目管理的基本概念
- 二、研发管理、IPD、CMMI、敏捷的定义及关系
- 三、软件开发模型与工作方法
- 四、研发项目管理及组织
- 五、研发项目计划制定
- 六、研发项目计划控制
- 七、在KD研发中心的某一天
- 八、研发项目精细化管理**
- 九、KD研发中心项目管理体系

微软项目管理精细化工具Checklist

- 进行源代码管理以及源代码检入的控制
- 进行定期的代码审查,分享经验和教训
- 每个开发人员都进行单元测试
- 自动进行每日构造,每天可以得到产品最新的版本
- 进行BVT测试(构造验证测试),在第一时间知道构造版本的质量
- 有统一的Bug管理工具,定期发送Bug报告
- 对有争议的Bug进行专家会诊
- 有统一的测试用例管理工具,定期发送测试用例报告
- 所有团队成员都可以从团队Portal上获取自己需要的文档,了解项目的状态
- 为整个项目团队以及开发、测试人员建立各个Email List,便于进行有效沟通

微软项目管理精细化工具Checklist

1. 进行源代码管理以及源代码检入的控制



微软项目管理精细化工具Checklist

2. 进行定期的代码审查, 分享经验和教训



配合项目经理, 对昨天发现的典型**BUG**及今天修改的内容组织 **code review**. 发现问题及时修正

微软项目管理精细化工具Checklist

3.每个开发人员都进行单元测试 ✓



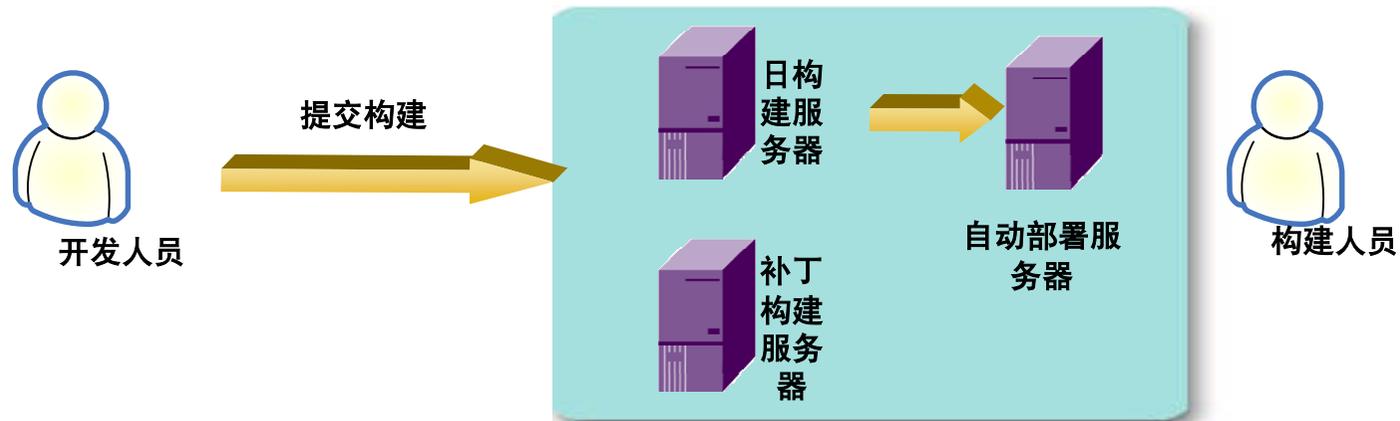
2 从Dailybuild获取最新环境, 将昨天已经开发完成的功能点, 执行单元测试, 通过后, 通过DMP流程提交需求验证

■ 常用测试工具:

- *COMUnit (自主开发; BOS平台开发插件测试工具)*
- *Nunit*
- *ACT (Microsoft Application Center Test) (并发压力测试)*

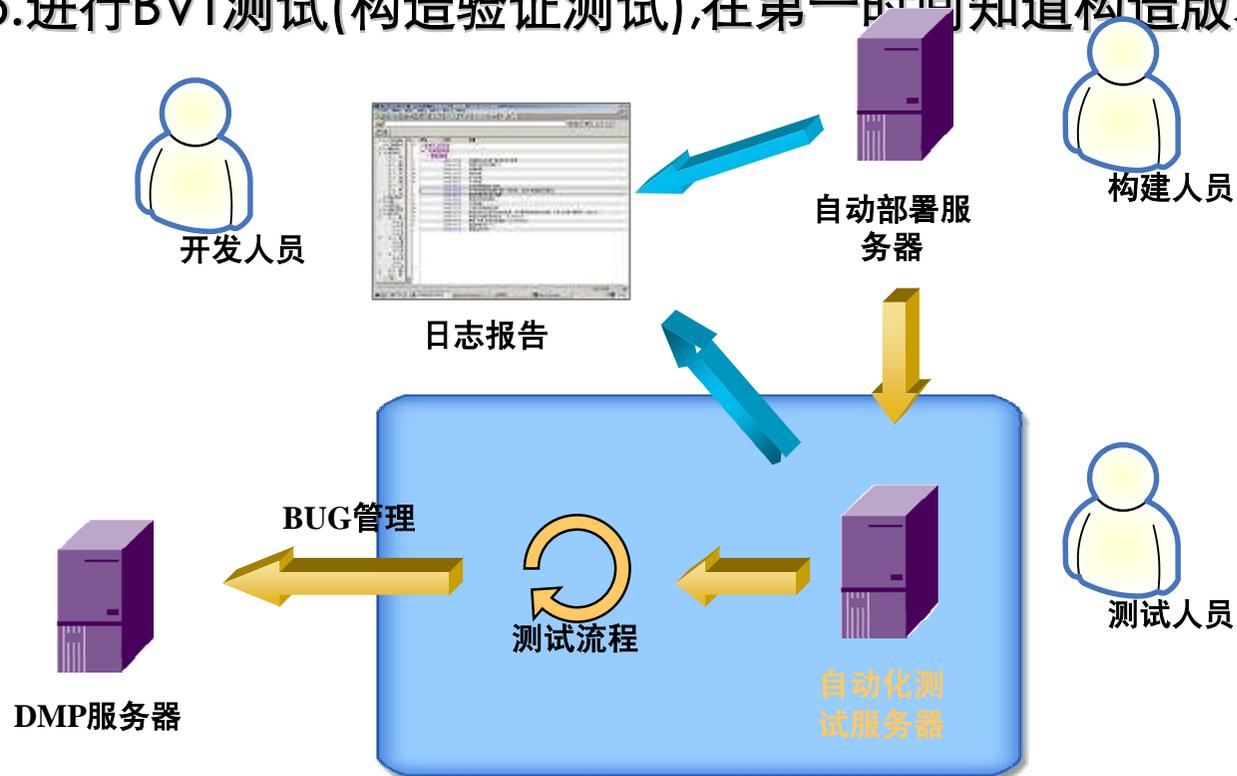
微软项目管理精细化工具Checklist

4. 自动进行每日构造, 每天可以得到产品最新的版本



微软项目管理精细化工具Checklist

5.进行BVT测试(构造验证测试),在第一时间知道构造版本的质量



微软项目管理精细化工具Checklist

6.有统一的Bug管理工具，定期发送Bug报告

Kingdee DMPV 2.3 用户：王文钦 用户名：jeff 所在部门：项目管理部

我的任务中心

- 我接收的任务
- 我发送的任务
- 工作日志
- 耗费中心
- 常用功能
 - 日常任务清单
 - 项目任务清单
 - 测试清单
 - 耗费录入
 - 加班登记
 - 补丁清单
 - 反馈录入
 - 反馈清单
 - 测试用例任务清单

新增 修改 确认失效 反失效 刷新 过滤 补充描述 提交 workflow workflow 处理

序号	测试记录代码	产品任务	产品名称	产品版本	子系统	项目组	迭代阶段	缺陷性质	引入阶段	缺陷类型
1	BT262691	K/3V11.0多	K/3	11.0.0	许可管理	K/3基础平台开	beta测试阶段	功能	需求	功能错误
2	BT262688	K/3V11.0多	K/3	11.0.0	许可管理	K/3基础平台开	beta测试阶段	功能	需求	功能错误
3	BT262684	K/3V11.0多	K/3	11.0.0	BOS IDE公用	K/3BOS开发组	beta测试阶段	功能	设计	功能错误
4	BT262682	K/3V11.0多	K/3	11.0.0	许可管理	K/3基础平台开	beta测试阶段	功能	需求	人机交互
5	BT262680	K/3V11.0多	K/3	11.0.0	总账	K/3财务开发组	beta测试阶段	功能	编码	功能错误
6	BT262647	K/3V11.0多	K/3	11.0.0	个人薪酬核算	K/3基础人事开	beta测试阶段	功能	需求	功能错误
7	BT262642	K/3V11.0多	K/3	11.0.0	应收应付	K/3应收应付开	beta测试阶段	易用性	集成	人机交互
8	BT262641	K/3V11.0多	K/3	11.0.0	合并账务	K/3财务开发组	beta测试阶段	易用性	集成	人机交互
9	BT262635	K/3V11.0多	K/3	11.0.0	网上结算	K/3成本开发组	beta测试阶段	易用性	集成	人机交互
10	BT262632	K/3V11.0多	K/3	11.0.0	职员管理	K/3人力发展开	beta测试阶段	易用性	集成	人机交互
11	BT262631	K/3V11.0多	K/3	11.0.0	销售管理	K/3供应链开发	beta测试阶段	易用性	编码	人机交互

问题详细描述：
英文环境下，【个人核算设置】→【类别管理】→【人事应用】，导入绩效数据，过滤条件已经设置了考核周期开始日期和结束日期，但是导入时，仍然提示导入前要输入开始和结束日期。

操作步骤：

图片：

数据环境：

工作流日志：

节点	节点名称	操作	对象状态	处理人	接收人	处理时间
0	提交	提交	提交	郭宇鹏	黄旭辉	2008-03-20 11:00
1	已读	提交	已改	黄旭辉	郭宇鹏	2008-03-20 11:00
3	已改验证	转发	已改	张建芳	吴永华	2008-03-24 11:00
3	已改验证	结束	验证通过	吴永华		2008-03-24 11:00

完成 可信站点 100%

微软项目管理精细化工具Checklist

7.对有争议的Bug进行专家会诊



主持Bug专家会诊等专题讨论

微软项目管理精细化工具Checklist

7.对有争议的Bug进行专家会诊-案例:BUG分析:

- 目的：坚持不懈进行BUG分析，提高开发人员的编码能力和质量意识，从而有效控制代码质量；
- 周期：功能测试和集成测试期间，每周组织一次；
- 方法：开发人员抽取典型BUG，形成BUG分析文档，组织部门会议，进行交流分享
- 要求：BUG分析文档必须描述以下内容：
 - 问题描述、分析过程、解决方法、思考启示



K3V11.0_Bug分析_
基础人事_蒙之新_20071010.



K3V11.0_HR开发部
_BUG分析交流会议总结_2007

微软项目管理精细化工具Checklist

8.有统一的测试用例管理工具，定期发送测试用例报告



测试用例设计

导入DMP管理

测试用例执行

The screenshot displays the Kingdee DMPV 2.3 software interface. On the left, a tree view shows project structure under 'All projects', including '11新产品测试组', '01测试用例', and '02财务'. The main window shows a '测试用例任务清单' (Test Case Task List) table with columns for '序号', '用例任务代码', '用例编号', '用例级别', '产品任务', '项目组', '子系统', '用例任务名称', '测试任务代码', and '测试任务名称'. Below the table, there are sections for '详细描述' (Detailed Description), '操作步骤' (Operation Steps), '操作日志' (Operation Log), and '对应BUG' (Corresponding Bugs).

序号	用例任务代码	用例编号	用例级别	产品任务	项目组	子系统	用例任务名称	测试任务代码	测试任务名称
1	TC086392	PRJ00086887_1	1	K/3V11.0多语 K/3财务调	合并报表	多语言维护_基础	PRJ00086887	多语言维护_基础	
2	TC086393	PRJ00086887_1	1	K/3V11.0多语 K/3财务调	合并报表	多语言维护_基础	PRJ00086887	多语言维护_基础	
3	TC086394	PRJ00086887_1	1	K/3V11.0多语 K/3财务调	合并报表	多语言维护_基础	PRJ00086887	多语言维护_基础	
4	TC086395	PRJ00086889_1	1	K/3V11.0多语 K/3财务调	合并报表	多语言维护_币别	PRJ00086889	多语言维护_币别	
5	TC086396	PRJ00086889_1	1	K/3V11.0多语 K/3财务调	合并报表	多语言维护_币别	PRJ00086889	多语言维护_币别	
6	TC086397	PRJ00086889_1	1	K/3V11.0多语 K/3财务调	合并报表	多语言维护_币别	PRJ00086889	多语言维护_币别	

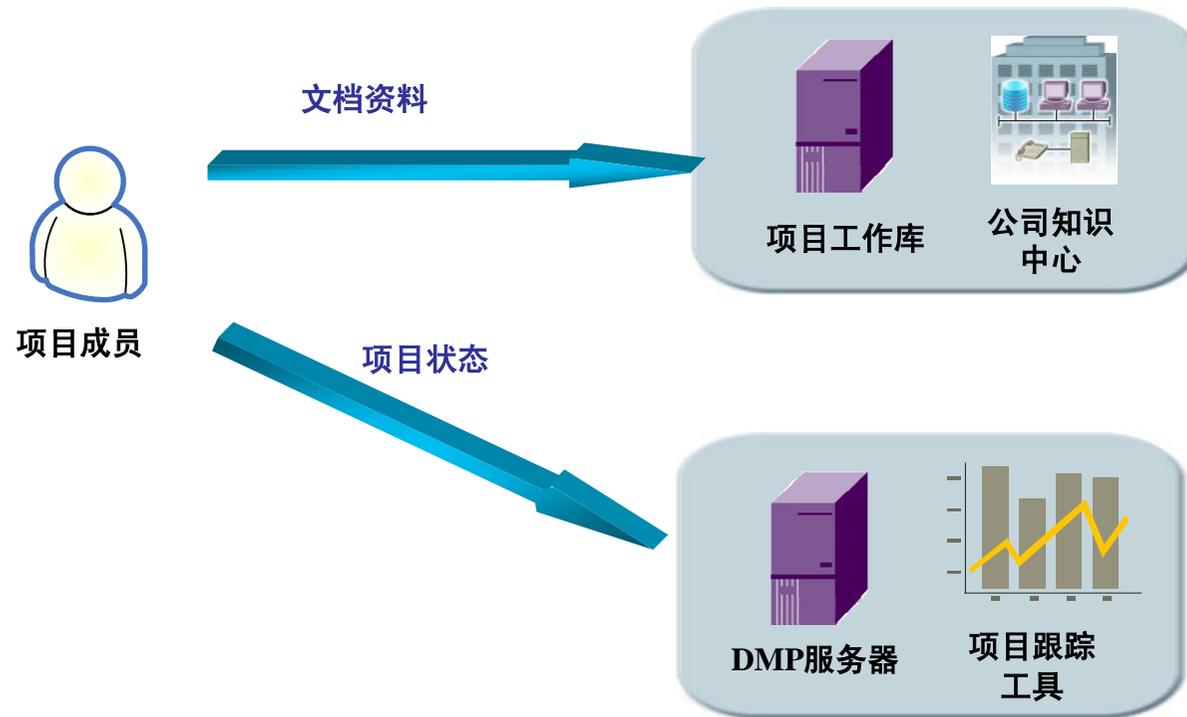
序号	操作	操作人	时间	操作说明
1	用例标注	系统管理员	2008-01-1	简体: 不通过
2	用例标注	系统管理员	2008-01-1	简体: 通过 英文: 通过
3	用例标注	刘春燕	2008-01-1	简体: 通过 繁体: 通过
4	用例标注	系统管理员	2008-01-1	英文: 通过

序号	BUG代码	BUG关键字	BUG状态
1	BT255145	股权清单新增, 界面排列	验证通过



微软项目管理精细化工具Checklist

9.所有团队成员都可以从团队Portal上获取自己需要的文档,了解项目的状态 ✓

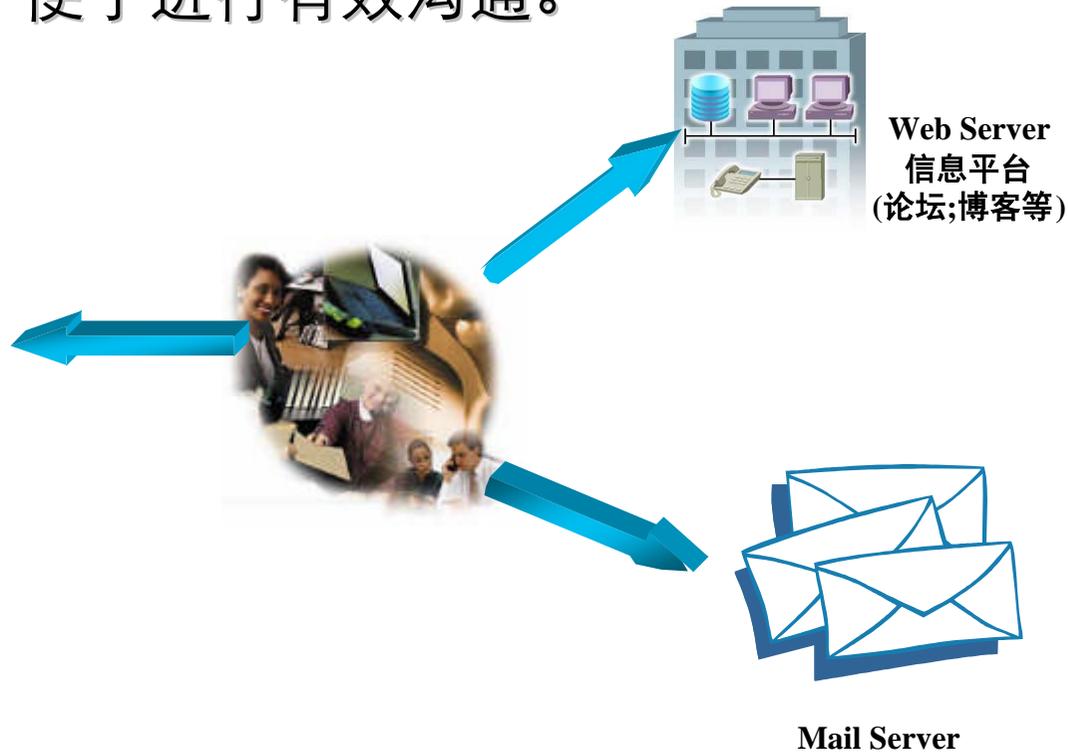


微软项目管理精细化工具Checklist

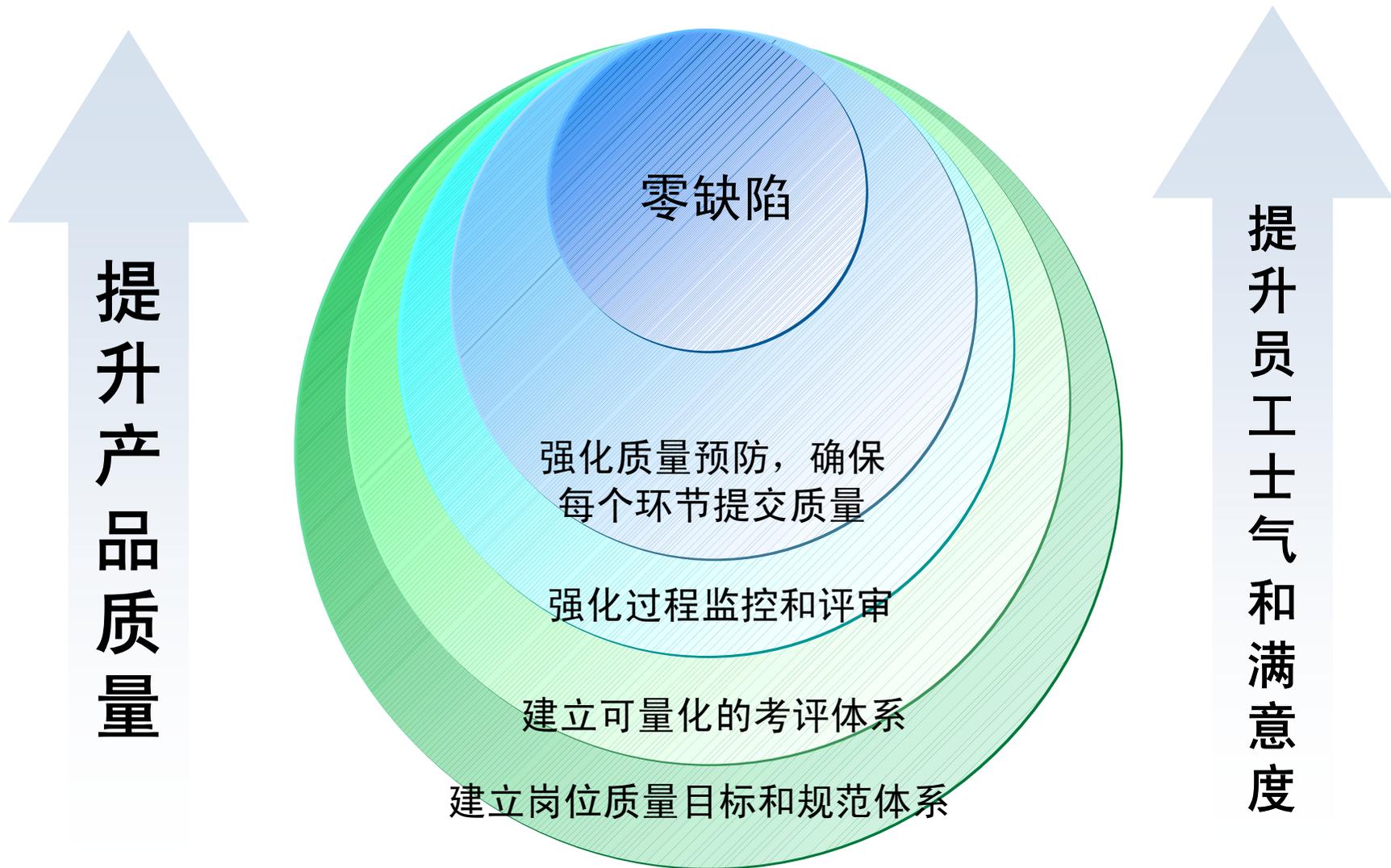
10. 为整个项目团队以及开发、测试人员建立各个 Email List，便于进行有效沟通。



RTX Server



在软件项目精细化管理的探索和应用





DINGLI COMMUNICATIONS CORP.,LTD.



目录 *Content*

- 一、研发项目与项目管理的基本概念
- 二、研发管理、IPD、CMMI、敏捷的定义及关系
- 三、软件开发模型与工作方法
- 四、研发项目管理及组织
- 五、研发项目计划制定
- 六、研发项目计划控制
- 七、在KD研发中心的某一天
- 八、研发项目精细化管理
- 九、KD研发中心项目管理体系

KD项目管理体系简介

KD项目管理工具

工作文档配置管理: VSS

开发配置管理:

- 自动编译工具;补丁构建工具; VSS; CVS;
- 微软成功秘诀之一:

- 自动发布服务器

- Daily Builder

测试部署: 自动化部署工具; 自动化测试

项目管理工具:

- DMP (Development Manager Platform)
- RMP (Requirement Manager Platform)



KD项目管理体系简介

KD项目管理方法

标准化流程和相应规范

- 协作的依据, 和基本质量保障

标准化和平台化

- 两大管理平台: DMP; RMP
- 六大工作流程
- 八大服务器
- 耗费管理
- 通过耗费数据跟踪和分析项目进度和质量

过程跟踪

- 规范度和进度/质量的指标量化

测试系统化

- 单元代码测试->功能测试->集成测试->金盘测试->BETA测试
- 每个阶段都有明确的可量化的指标要求

日构建 / 自动化测试



KD项目管理体系简介

KD项目管理工具-六大流程

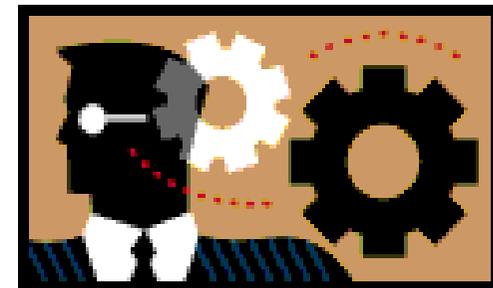


P148

KD项目管理体系简介

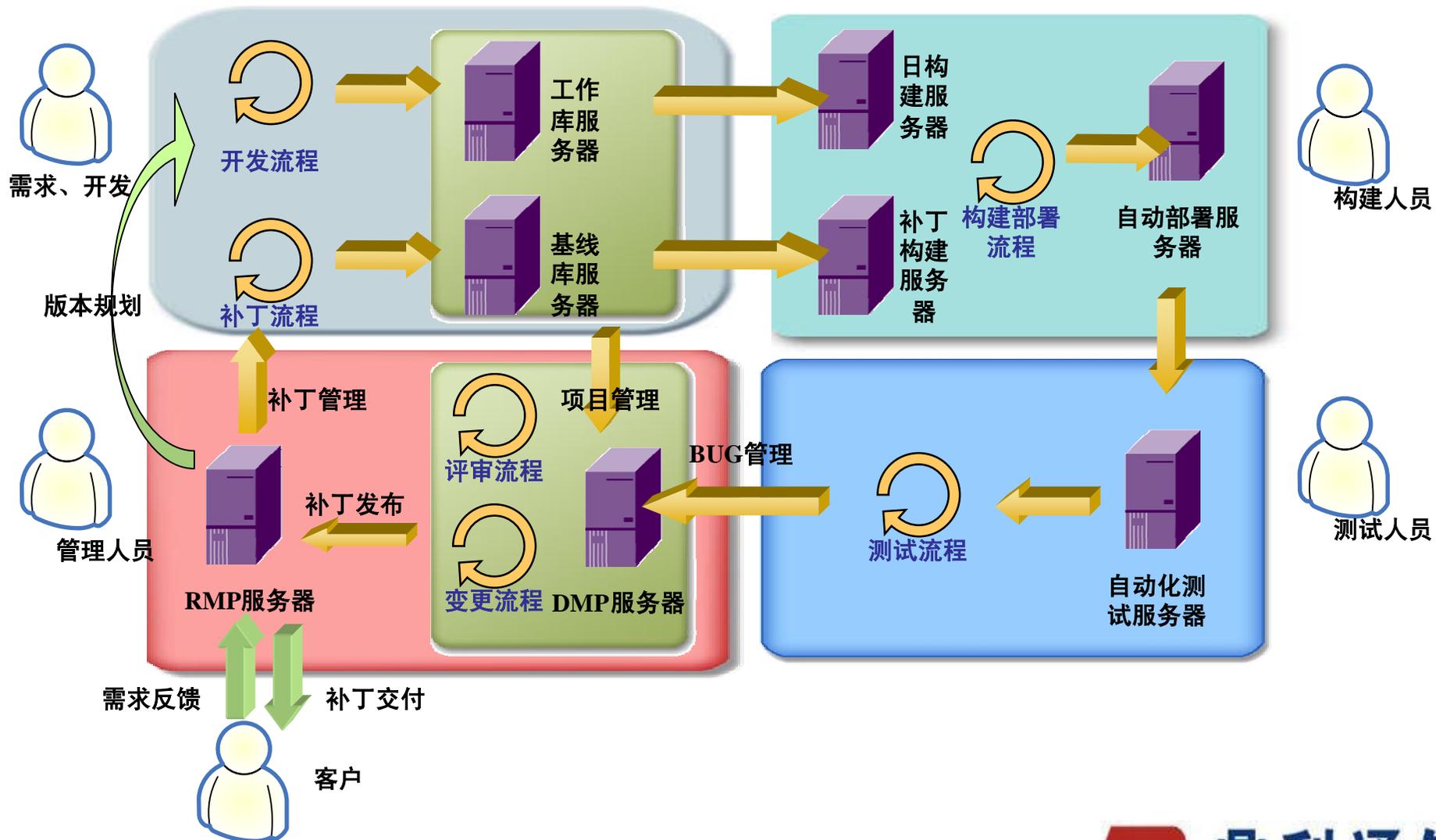
KD项目管理工具-八大服务器

- 项目管理平台(DMP)服务器(项目任务; 需求; 测试; 补丁; 评审; 变更; 协同)
- 需求反馈管理平台(RMP)服务器
- 工作产品服务器(工作库/基线库)
- 代码服务器(工作库/基线库)
- 日构建服务器
- 补丁构建服务器
- 自动化测试服务器
- 自动化部署服务器



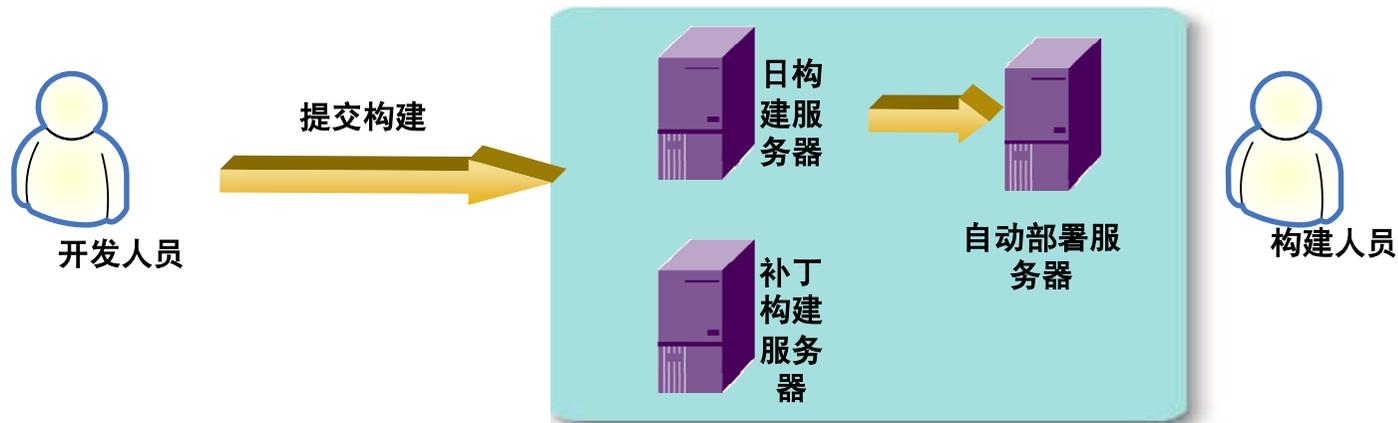
KD项目管理体系简介

KD项目管理工具-流程整合示意



KD项目管理体系简介

日构建的优势



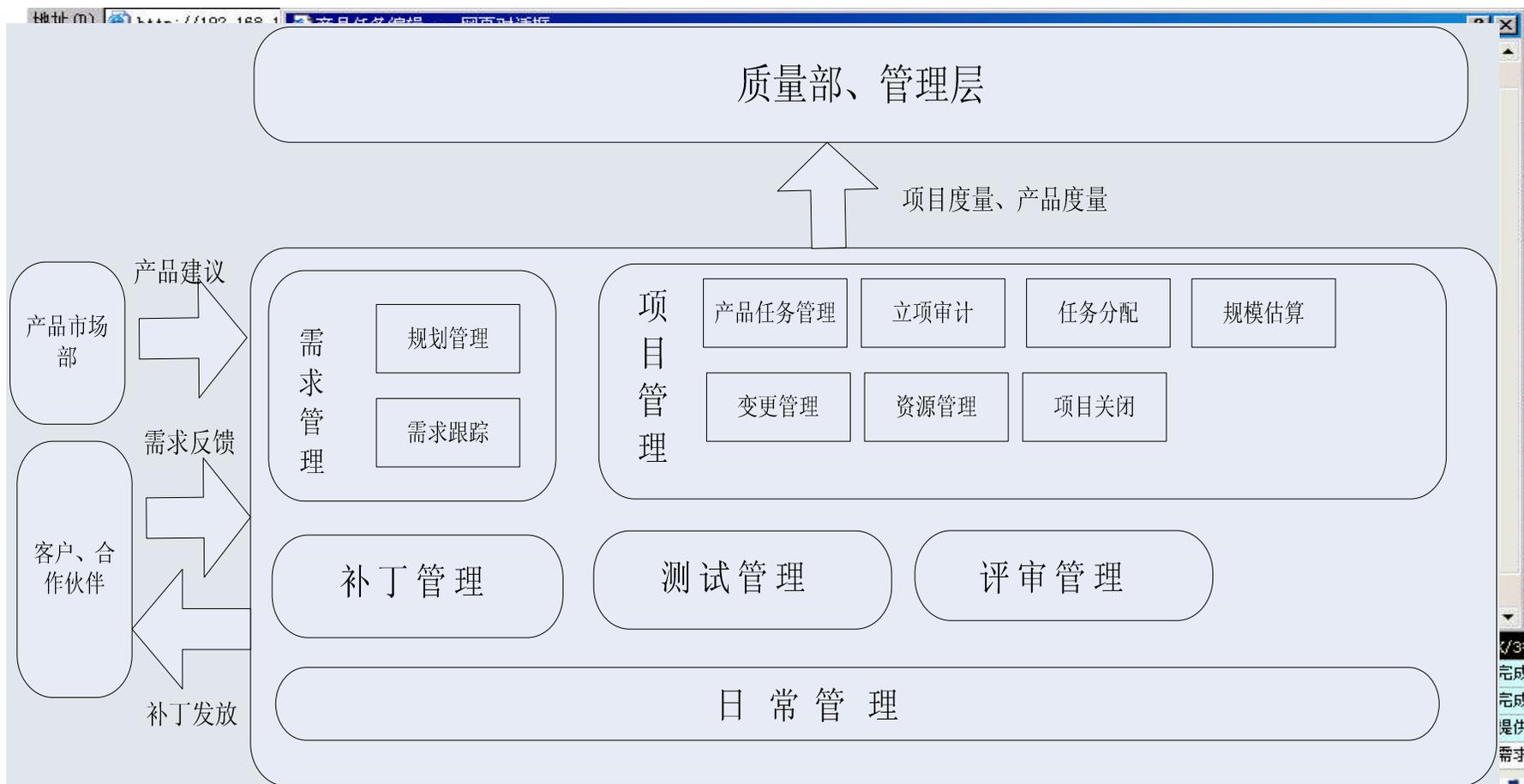
- 标准化组建编译打包和部署的工作流程，使得开发和测试每天都能获取最新的环境，提高了工作的协作效率
- 全自动化，提高工作效率

- 要求开发人员对代码和脚步的认真检查和自动化校验，提高了组件的交付质量和集成质量

KD项目管理体系简介

KD研发管理平台-DMP

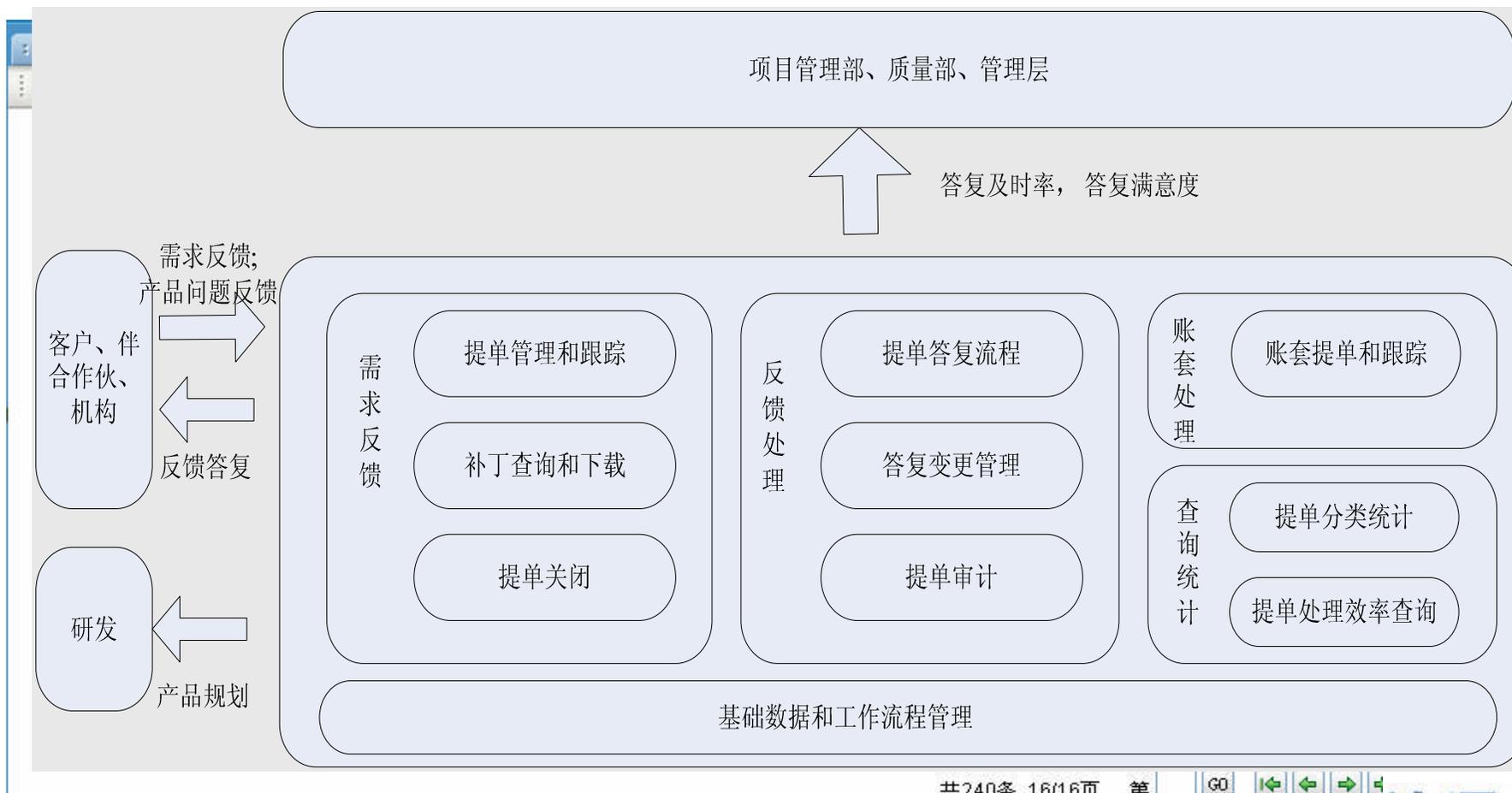
- 各类角色（规划、需求、开发、测试）协同工作平台
- 软件开发过程管理平台
- 质量指标数据收集分析平台



KD项目管理体系简介

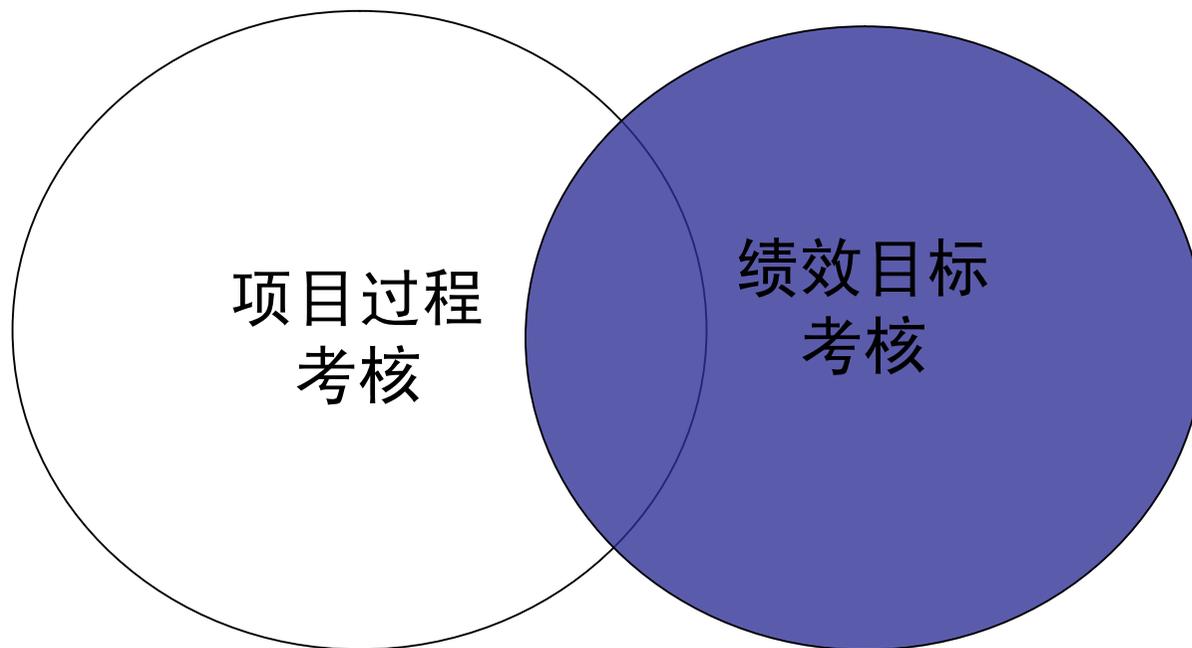
KD需求反馈管理平台-RMP

客户问题反馈服务支持的平台



KD项目管理体系简介

KD项目考核体系



KD项目管理体系简介

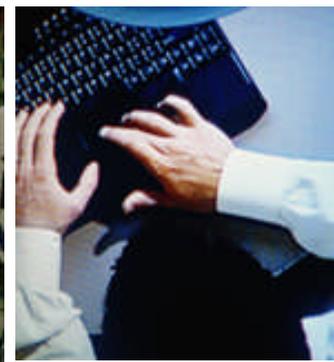
过程考核

产品审计

- FURPS指标
 - F: 功能性
 - U: 可用性
 - R: 可靠性
 - P: 性能
 - S: 可支持性
- 客户验证情况
- 全产品资料完成情况

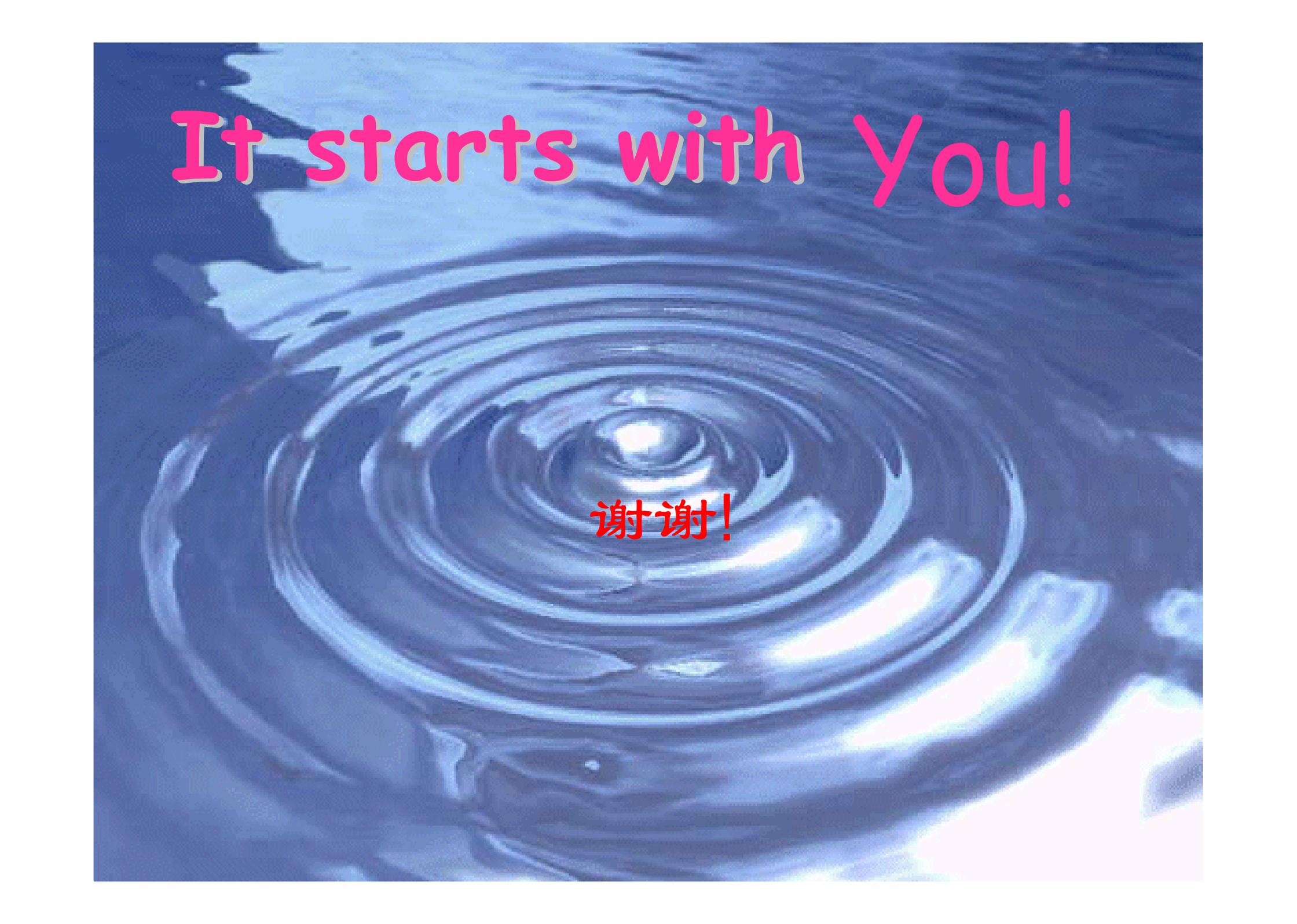


DINGLI COMMUNICATIONS CORP.,LTD.



目录 *Content*

- 一、研发项目与项目管理的基本概念
- 二、研发管理、IPD、CMMI、敏捷的定义及关系
- 三、软件开发模型与工作方法
- 四、研发项目管理及组织
- 五、研发项目计划制定
- 六、研发项目计划控制
- 七、在KD研发中心的某一天
- 八、研发项目精细化管理
- 九、KD研发中心项目管理体系



It starts with You!

谢谢!