



微软过程

- 1 微软过程概述
- 2 微软过程术语
- 3 微软过程基本原则
- 4 微软过程特点——MP与AP、RUP
- 5 微软过程实施策略
- 6 微软过程案例——IE浏览器经典战役
- 7 本章小结
- 习题

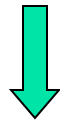
1 微软过程概述

■ 微软的成功

■ 世界上最大的、也是最成功的软件公司

产品涵盖编程工具（如VC、VB、.NET）、操作系统（如Windows、WindowCE）、数据库（如SQL Server）、网络产品（如IE、MSN）、办公产品（如Office）等多个领域

■ 未通过CMM认证，也未宣称使用过RUP或AP



■ 微软有他自己的软件开发过程

其过程被几十年实践证实是非常成功且行之有效

微软开发过程与RUP、敏捷过程有何区别、有无共同点？

1 微软过程概述

■ 微软解决方案框架

(Microsoft Solution Framework, MSF)

■ MSF的提出

微软顾问咨询部于1994年根据微软公司成功的产品开发经验总结、设计而成的框架体系

■ MSF发展史

微软将公司内部的产品开发人员、顾问咨询人员以及微软公司全球的客户和合作伙伴们在项目设计、开发和管理方面经过实践检验、可重复、可借鉴的成功经验都集成到了MSF之中

1 微软过程概述

■ MSF设计目的

- 帮助企业提升利用IT技术解决商务问题的能力
- 为不同规模的组织结构和不同类型的IT项目提供从项目组织规划和产品发布管理的全方位的指导和帮助
- 是一套高效、灵活、可扩展的软件开发管理体系

1 微软过程概述

- MSF内容： 框架结构的经验知识库

- 企业结构设计方案

- 采用交互的方式，侧重于制定长期规划，同时也能完成短期目标

- 项目开发准则

- 包括组队模型和过程模型，用于建立高效的项目组，管理项目组的生命周期

- 应用程序模型

- 用于支持设计复杂的分布式企业应用

- 企业信息基础设施的实施方案

- 使用组队模型和过程模型支持实现、操作和技术上的方案

1 微软过程概述

- 微软过程 (MP, Microsoft Process)

- 从MSF中抽取出项目开发准则中的过程模型和组织模型

- 构成了一套软件过程模式

内容涵盖软件过程中的过程、人员及组织、方法、产品等不同方面

2 微软过程术语

1. 项目前景 (Vision) 与项目范围 (Scope)

- **项目前景**是对项目要解决什么问题的开放性描述，它代表项目的远景目标
- **项目范围**描述的则是在项目的限制条件内，需要完成哪些具体的目标，这主要是指所有特定的近期目标而言

2 微软过程术语

2. 功能说明书

■ 内容

功能说明书阐释了软件每一个特性的功能和执行方式，以及所有特性的组合关系和整体架构

■ 形式：包括单页和详细两种形式

- 首先编制单页功能说明书，它概要性的描述所有产品特性的功能、性能及其在项目中的优先级，通常该说明书占一页纸的篇幅，需要2到5天的时间就可以完成
- 在此基础上编制详细功能说明书，从技术细节上详细描述如何实现所有的产品特性，该说明书占用的篇幅较长，一般需要1~3个月的时间来编写

2 微软过程术语

■ 用途

- 可以与客户确认要开发的究竟是什么样的软件
- 是项目计划的标准和依据
- 可以告诉开发人员将要开发的是什么样的软件
- 是所有项目组成员同步各自工作的基石

2 微软过程术语

3. 程序经理 (Program Manager)

- 程序经理的职责是在规定的项目资源、期限等限制条件下，确保产品能够如期发布，为此程序经理需要制定和管理项目日程、费用预算、产品特性说明书等文档，确保项目发起人的意图得到落实，并负责推动产品开发过程顺利进行，确保项目组可以在合适的时间交付合适的产品

- **程序经理**不同于传统的**项目经理**

在微软的团队组织结构中，六个组队角色的地位是相互平行、相辅相成，程序经理只是项目开发过程的组织者、管理者和决策者，不是项目的领导者

3 微软过程基本原则

从MSF的立场观点，论述MP

- 过程原则
- 组队原则

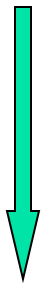
3 微软过程基本原则

- ■ MP的过程原则
- MP的组队原则

3.1 MP的过程原则

1. 制定计划时兼顾未来的不确定因素

■ 任何项目都包含不确定因素，如



- 需求可能不断变化
- 技术可能不断变化
- 市场环境可能变化

■ 考虑到未来可能发生的不确定因素

制定项目计划、进度表时，为不可预期的项目变更及项目风险留出一定的余地

点评：这一原则与AP第4条价值观“响应变化胜过遵循计划”异曲同工

3.1 MP的过程原则

2. 通过有效的风险管理减少不确定因素的影响

有效管理和控制不确定因素的最好方法——
使用成熟的风险管理模型

点评：对照而言，RUP提出的风险管理方法为在每次迭代中都要解决最突出的风险问题，两者互补

3.1 MP的过程原则

3. 经常生成过渡版本并进行快速测试来提高产品的稳定性及可预测性

■每日生成制度（Daily Build）

- 在项目开发早期就会创建可以编译、生成的代码框架，并对每个项目组成员的代码进行同步处理
- 每天，开发工作结束前，项目组成员都会将一天中对代码所做的修改检入到版本库中
- 所有的代码检入操作之后都要生成一个完整地、可执行的产品版本，并对生成结果进行快速测试——这种测试通常被称为“生成验证测试（Build Verification Test, BVT）”
- 当日的修改中可能出现的、影响到产品编译、连接的Bug就可以被及时发现并及时排除，每天工作结束后，项目组总能提供一套可执行的过渡产品版本

点评：在最大程度上保证整个产品开发过程可管理、可预期，并能增强产品的稳定性，类似AP的持续集成

Square-Ca13.0失败的主要原因

3.1 MP的过程原则

快速循环、递进的开发过程

要求项目组在开发过程中迅速完成每一次递进过程，并在每一个开发周期中都能切实地增加产品特性，提高产品质量

点评：和敏捷过程所强调的不断重复产品的生命周期、以递进的方式推出版本的要求相似

3.1 MP的过程原则

5. 从产品特性和成本控制出发创造性地工作

- 时刻关注产品特性的开发和项目资源的控制之间的平衡

因为任何商业软件开发项目的最终目标都是以有限的成本实现所有客户需要的产品特性

- 微软提倡“聪明地工作”

项目组并不对其成员每天的工作时间作硬性的规定，而是要求其成员能够创造性地工作

3.1 MP的过程原则

6. 创建确定的进度表

- 项目进度表应当具有较高的稳定性和权威性

项目组的各项工作都应当按照进度表的要求开展。在项目过程中随意对项目进度表进行修改和变更是不明智的做法，这只会导致项目组内、外项目过程和项目目标的困惑

- 必须调整项目进度的时候，项目组也应当遵照严格的变更管理程序完成进度调整工作

点评：在具体制定项目进度表方面，借鉴AP的策略，即制定一种细致度逐渐降低的进度计划以保持足够的灵活性

3.1 MP的过程原则

7. 使用小型项目组并发完成工作，并设置多个同步点

- 微软公司的项目组善于将较大的项目分解成多个子项目
- 由多个小型项目组并发地完成工作
- 要在项目过程中设置多个同步点（Synchronization point）
 - 保证所有小型项目组之间工作目标的统一和工作任务的同步
 - 里程碑是最为重要的项目同步点

3.1 MP的过程原则

8. 将大型项目分解成多个可管理的单元，以便更快地发布产品

- 大型项目分解成不同的产品单元的拆分方法：

 - 根据产品特性

- 每一个产品单元：

 - 拥有自己特定的工作目标并由一个小型项目组负责

→ 将有效缩短产品的发布周期

3.1 MP的过程原则

9. 用产品的前景目标和概要说明指导项目开发工作——先基线化，后冻结

■ 先基线化

- 产品的前景目标和概要性说明在项目的全过程中都应当是明确和稳定的，不经过严格的变更管理程序，任何人都无权修改这些文档中的关键性内容
- 对于源代码、设计文档、用户手册等配置项，一般在项目早期为其建立明确的、可度量的基线；在项目过程中，项目组可以根据需要，通过变更管理程序对基线库中的各配置项进行修改和跟踪

■ 后冻结

产品冻结发生在项目过程后期的稳定和测试阶段，基线库中的配置项一经冻结，就不能再随意更改，对配置项的冻结有利于与产品发布相关的工作顺利进行

点评：冻结思想与AP不同，AP倡导即使到了开发的后期，也欢迎改变需求

3.1 MP的过程原则

10. 避免产品走形

检查和审视当前项目状态、产品特性是否和产品的功能说明书相吻合

点评：对照而言，RUP中避免产品走形的方法是用例驱动

3.1 MP的过程原则

11. 使用原型验证概念，进行开发前的测试

原型验证概念（proof-of-concept prototyping）

- 内容：在产品开发工作开始之前，项目组应使用原型对产品需求、技术可行性和项目范围进行早期论证
- 作用：这种开发前的测试可以有效减少不确定因素的影响，避免项目风险

3.1 MP的过程原则

12. 零缺陷观念

零缺陷（zero-defect）并不意味着产品中没有任何Bug

- 首先按零缺陷这一高标准进行要求
- 具体实施时，项目组在产品的每一个阶段、在发布产品的每一个版本之前，都对已发现的产品Bug进行了有效的管理和控制，改正了影响产品使用的Bug，对不影响产品使用、且因资源有限无法及时修改的Bug进行跟踪和记录，确保产品中所有已发现Bug都在项目组的控制范围之内，都可以在适当的时机得到修正

3.1 MP的过程原则

13. 非责难式的里程碑评审会

里程碑评审会——质量保证会议

- 会议主旨不在于追究项目组或项目组成员的责任
- 目的发现项目中存在的问题并及时解决问题
讨论项目中哪一些产品特性没有完成、哪一个阶段工作延期了、为什么没有完成、为什么延期、是否需要再追加投入、是否需要调整项目组的人员结构等等
- 比尔⑩盖茨也经常参加重要产品的评审会

3.1 MP的过程原则

■举例：

IE项目组没能及时推出IE的Beta2测试版，他们召开工作会议对这一情况进行考察和讨论。在会上发现，大多数小项目组都按时完成了工作，只有一个项目组例外，但是大家没有简单地责怪他们，而是一起帮助这个项目组寻找工作延期的原因。后来发现，这个项目组无法按时完成开发工作的原因是他们负责的产品特性比预计的要复杂得多，即使换了其它项目组也没有办法在预定的时间里完成所有开发工作。找到了问题的根源之后，项目组采取了相应的解决办法：为该项目组补充了人员，并砍掉了一些次要的产品特性。这样一来，项目组就可以很好地完成产品开发工作了

点评：MP的里程碑与RUP的里程碑的思想几乎是一致的

3 微软过程基本原则

- MP的过程原则

→ ■ MP的组队原则

3.2 MP的组队原则

1. 小型的、多元化的项目组

■ 小型化项目组

■ 意义

只有在小型的、拥有确定目标和预算的项目组中，项目组成员才能更好地分工协作、更好地发挥个人在技术或管理上的经验和技能（比尔·盖茨语）

- 交流成本、运营成本和管理成本低
- 决策和执行速度快
- 产品质量高

点评： AP鼓励的也是一种小型化的项目组

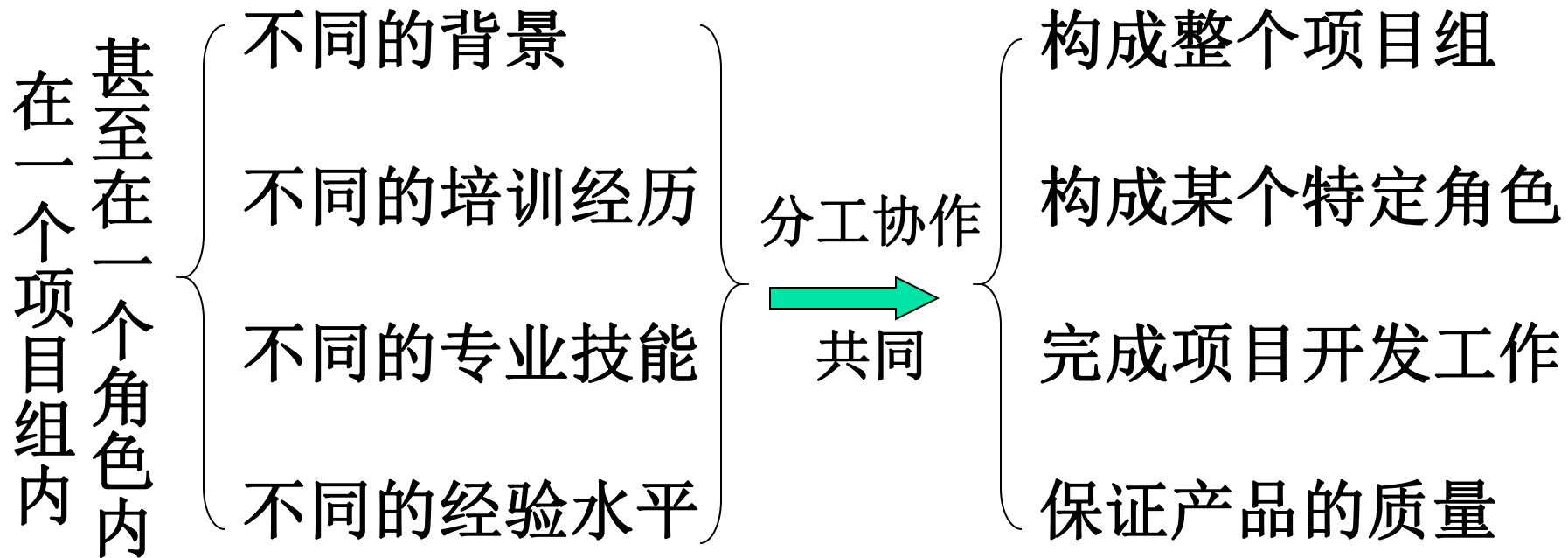
3.2 MP的组队原则

■ 小型化项目组

- 举例1：一个程序经理需要管理40个项目成员
 - 为保持组内有效沟通，每周要与每一个项目组成员进行至少一小时的单独谈话，那么，他就没有任何时间处理其他事情了
 - 不同的人有不同的个性和不同的观点，很难把许多人集中在一个项目中，统一安排工作
 - 相反，如果把40个人按照不同的层次结构或职能单位划分成几个小的项目组，每个小项目组大约5个人，每个人的职责划分就会清晰，更沟通和交流、控制每个人的开发质量和进度
- 举例2：项目组交流成本 $N * (N - 1) / 2$

3.2 MP的组队原则

■团队多元化的体现



3.2 MP的组队原则

2. 角色依赖和职责共享

■ 原因

- 无法将项目中不同角色的工作完全孤立开来
- 如果每个角色都对整个项目蓝图有一个清晰的认识，项目组的效率就会成倍提高
- 相互依赖的工作促使所有项目组成员在他们直接负责的领域之外主动发表意见、贡献力量 \Rightarrow 提高项目组内的知识、技能和经验的共享程度

■ 实施：对等团队（Team of Peers）结构

点评： 与AP第11条原则中提出的“最好的构架、需求、设计出自于自组织的团队”思想基本吻合

3.2 MP的组队原则

3. 专深的技术水平和业务技能

■ 相关领域的业务技能

■ 要求：

- 熟悉客户的业务流程、业务模式
- 透彻地了解客户需求

■ 意义：

- 把握住产品的设计目标
- 开发出可以令客户满意的产品

3.2 MP的组队原则

■ 专深的技术水平

■ 要求：

- 熟练掌握相关技术
- 及时跟踪相关领域的技术发展情况

■ 意义：

- 使用最合适的技术进行产品开发
- 确保最终产品的性能和质量

3.2 MP的组队原则

以产品发布为中心

- 产品发布意识的内涵

- 在市场上或在公司内部发布软件产品
- 在更高的层面上，产品意识要求成员将自己每一次劳动的成果都看成是自己贡献给整个团队的一件产品

- 培养产品发布意识

例如为每一个产品给出一个显著的标识，如

- 赋予每个产品或每个产品单元一个内部代码
 - 有助于明确产品来源，考察项目组工作，增强项目组成员的责任心，显著提高项目组士气
- 把产品代码印在T恤衫、咖啡杯或者其它小礼品上
 - 有效提高项目组的自我认同感

3.2 MP的组队原则

5. 明确的目标

- 明确的目标是项目协同工作的依据
- 明确的目标还必须与客户的最终业务需求相吻合

3.2 MP的组队原则

6. 客户的主动参与

参与方式包括

- 邀请客户参与产品的设计，并在产品开发过程中随时征询客户的反馈意见
- 设置产品管理这一角色：以客户的身分向项目组提出业务需求，有时产品管理角色中的某些成员甚至是由客户直接担任的

点评：

- 与AP中强调客户这一角色重要性的目的是一致的
- 微软设置产品管理角色方法相对AP的与客户天天工作在一起的方法更具可实现性

3.2 MP的组队原则

7. 分享产品的前景

■ 分享产品前景的积极作用

- 以自己能为产品的美好前景贡献力量而自豪
- 每一位成员都在产品前景的激励下努力工作

■ 不能分享产品前景的消极影响

- 对工作目标产生困惑或迷茫
 - 对项目前景产生自己的看法
 - 每一个成员判断自己工作是否成功的标准也会大相径庭
- ➡ 无法调动积极性、无法切实保证产品开发的顺利进行

3.2 MP的组队原则

8. 所有人都参与设计

- 不同的角色、不同的成员对产品的设计有着不同的视角和看法
- 集思广益有助于设计出的产品更加趋于完善和合理

点评：与AP第11条原则“最好的构架、需求和设计出自于自组织的用户”是一致的

3.2 MP的组队原则

9. 认真从过去的项目中吸取经验

没有哪一个项目组可以永远成功

- 善于从成功的项目中总结项目管理的诀窍
- 勇于对失败的项目进行分析和反省

→ 项目组才能不断进步

3.2 MP的组队原则

10. 共同管理、共同决策

- 每个成员应当积极参与项目组中每一个重要的决策过程

尽管每一个团队成员的职责可能都不尽相同，但每个成员都对项目中的重要决策负有一定的责任

- 项目组中每一个角色的负责人仍然拥有该职责领域的最终决定权

共同管理和共同决策并不意味着项目组中的所有人都可以不受限制地对项目组的工作安排指手画脚

3.2 MP的组队原则

11. 项目组成员在同一地点办公

- 过去成功和失败的项目管理经验表明，在同一个办公地点工作的项目组有着更高的沟通效率和更好的工作业绩

尽管今天通信技术的完善，电话和电子邮件相互沟通工作在不同的地点的项目组的各成员

- 工作在同一个楼层或同一个办公室，项目成员之间就会有相当多的机会进行非正式的交流

人际关系因此而得到改善，项目中很多棘手的问题也可以在电梯间、午餐桌等非正式的场所里得到解决

3.2 MP的组队原则

12. 大型项目组也像小型项目组一样运转

- 在项目组成立之初将较大的项目团队按产品特性拆分成若干个小项目组
- 这些小型项目组按照MSF组队模型进行管理和角色划分，并对各自工作目标负责
- 小型项目组之间通常是并行的工作关系，各项目组之间设置同步点

4 微软过程特点——MP与AP、RUP

从软件过程模式的四要素及其相互关系的角度，对比RUP、AP，分类分析总结微软过程这一软件过程模式的特点

- 生命周期
- 人员
- 方法
- 产品
- 相互关系

4 微软过程特点——MP与AP、RUP

→ ■MP的生命周期

■MP的人员

■MP的方法

■MP的产品

■MP的相互关系

1 MP的生命周期

1. 生命周期描述

■ 微软过程：

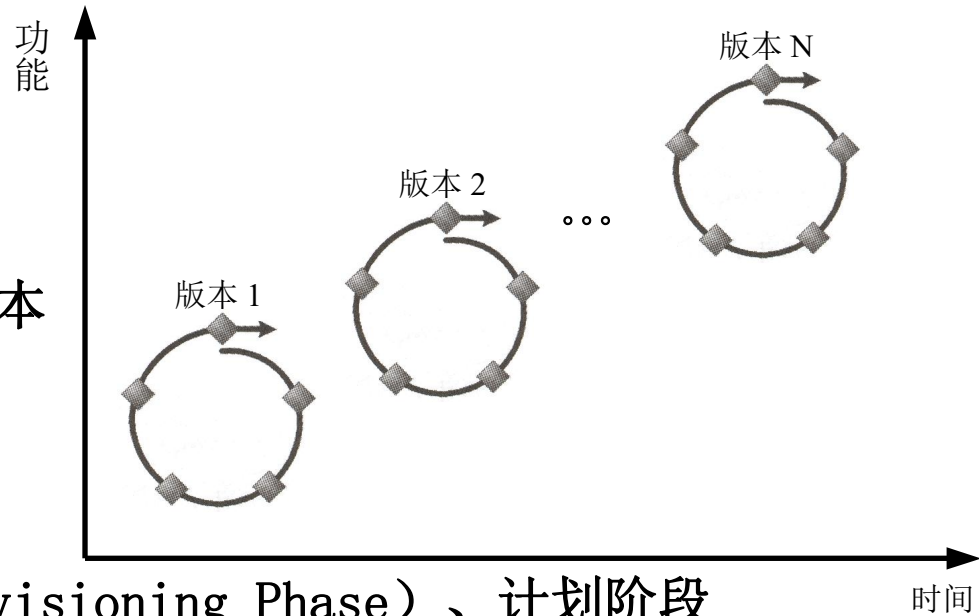
- 每个循环发布一个递进版本
- 各循环持续快速进行

■ 每个生命周期循环：

分为五个阶段：构想阶段（Envisioning Phase）、计划阶段（Planning Phase）、开发阶段（Developing Phase）、稳定阶段（Stabilizing phase）和发布阶段（Deploying Phase）

■ 每个阶段：

- 涉及产品管理、程序管理、开发、测试、发布各角色及其活动
- 各阶段结束于一个重要里程碑，阶段之间具有缓冲时间



1 MP的生命周期

生命周期中的阶段	主要里程碑	应提交的产品项
第一阶段——构想	前景/范围得到认可	<ul style="list-style-type: none">● 前景/范围说明书● 风险评估说明书● 项目组织结构说明书
第二阶段——计划	项目计划得到认可	<ul style="list-style-type: none">● 功能说明书（包括单页功能说明书和详细功能说明书）● 风险管理计划● 项目总体计划书和总体进度表
第三阶段——开发	项目范围内的所有产品特性开发完成	<ul style="list-style-type: none">● 源代码和可执行程序● 安装脚本和用于发布的配置信息● 已冻结的功能说明书● 关于产品使用的支持要素● 测试说明书和测试用例

1 MP的生命周期

第四阶段——稳定 可发布版本准备就绪

- 黄金版本
- 版本注释
- 关于产品使用的支持要素
- 测试结果和测试工具
- 源代码和可执行程序
- 项目文档
- 里程碑评审记录

第五阶段——发布 发布完成

- 运营与支持信息系统
 - 程序和过程
 - 知识库、报告、日志
 - 文档库：包含项目过程中产生的所有版本的文档、资源
 - 所有项目文档的最终版本
 - 下一步的工作计划
-
-

1 MP的生命周期

2. 生命周期特点

- 相对RUP，MP可视为RUP的一个精简配置版本
 - 整个过程：由若干生命周期持续递进循环
 - 每个循环：由若干阶段组成，且各阶段之间扩充为具有缓冲时间，各阶段的对应关系为
 - 先启阶段 ↔ 完成构想
 - 精化阶段 ↔ 完成计划
 - 构建阶段 ↔ 完成开发和稳定
 - 产品化阶段 ↔ 完成发布

1 MP的生命周期

- 每个阶段：

精简为一次迭代，每次迭代经历若干个工作流程
具体

- 先启阶段：主要经历业务建模、需求、项目管理
- 精化阶段：主要经历业务建模、需求、分析设计、项目管理
- 构建阶段：主要经历需求、分析设计、实现、测试
- 产品化阶段：主要经历部署、配置变更管理和项目管理

- 相对AP，MP是前者的一个扩充版本，扩充了其每个生命周期内的各阶段的具体运作流程

4 微软过程特点——MP与AP、RUP

- MP的生命周期

- ■MP的人员

- MP的方法

- MP的产品

- MP的相互关系

2 MP的人员

1. 人员及组织描述

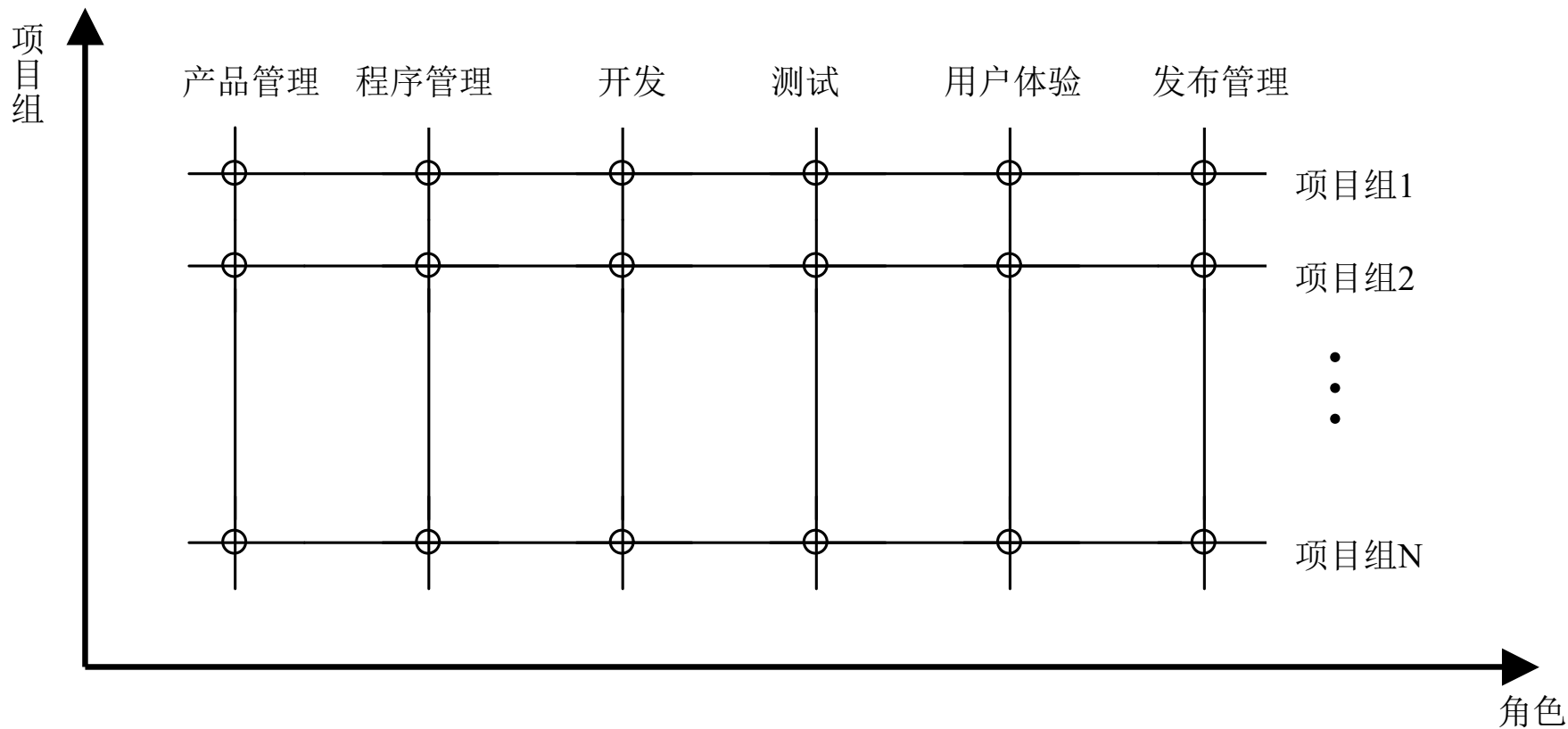
■ 人员分工

职责与任务分配类似按照RUP中的“角色”概念进行

■ 人员组织管理：矩阵结构

- 横轴：是按不同的专业技能划分的角色
- 纵轴：是由来自不同角色组中的各个角色组成的项目组

2 MP的人员



2 MP的人员

(1) 角色划分

- 项目组中职能的划分：六种角色
 - 产品管理角色
 - 程序管理角色
 - 开发角色
 - 测试角色
 - 用户体验角色
 - 发布管理角色

2 MP的人员

■ 六种角色职能活动表

角色	完成者	工作目标	应提交的产品项
产品管理	产品经理	提高客户满意度	<ul style="list-style-type: none">● 充当临时客户● 确保项目组成员对项目前景和项目范围了如指掌● 管理客户的需求定义● 开发、管理和提供业务用例说明● 管理客户的预期目标● 控制产品特性和开发周期的关系● 管理市场宣传和公共关系
程序管理	程序经理	在规定的项目资源、期限等限制条件下,确保产品能够如期发布	<ul style="list-style-type: none">● 推动产品开发过程● 管理产品范围和产品特性说明● 推动项目组内的交流和讨论● 管理产品开发进度、汇报项目状态● 控制项目开发中关键问题的取舍和决策
开发	开发工程师	使用适当的技术和工具实现项目目标、满足客户需求	<ul style="list-style-type: none">● 完成产品特性的物理设计● 在项目组内承担技术顾问的职责● 确保每一个产品特性在的的时间内完成● 使产品达到可发布的状态

2 MP的人员

测试	测试工程师	在产品最终发布前找到尽可能多的缺陷或错误	<div>1. 制定测试策略和测试计划</div> <div>2. 确保产品的所有特性都经过了严格的测试</div> <div>3. 向项目组提供翔实、准确的测试报告</div>
用户体验	用户体验人员	协助用户更好的使用产品，排除用户在使用产品时遇到的问题和障碍	<div>1 在产品阶段确保产品可被最终用户接受</div> <div>2 对产品的国际化功能提供支持</div> <div>3 设计和开发产品的技术支持系统</div> <div>4 用户培训</div> <div>5 确保产品的可用性</div> <div>6 图形用户界面设计</div>
发布管理	发布管理人员	确保产品的顺利发布，为项目组的正常运营提供支持	<div>1 代表项目组协调公司内的运营、支持、发布渠道等部门的工作</div> <div>2 项目组的后勤和基础设施管理</div> <div>3 管理产品发布事宜</div> <div>4 参与管理并支持相关的项目决策过程</div> <div>5 管理产品的认证或许可模式</div>

2 MP的人员

■ 最具特色角色——程序管理和产品管理

■ 诞生史

- 最初近十年未程序经理这一职

微软公司成立于1975年，但直到1984年才设置程序经理这一职位

- 这十年间，公司规模不断扩大，软件项目的复杂度与日俱增

→ 导致传统软件开发方式暴露出诸多弊端

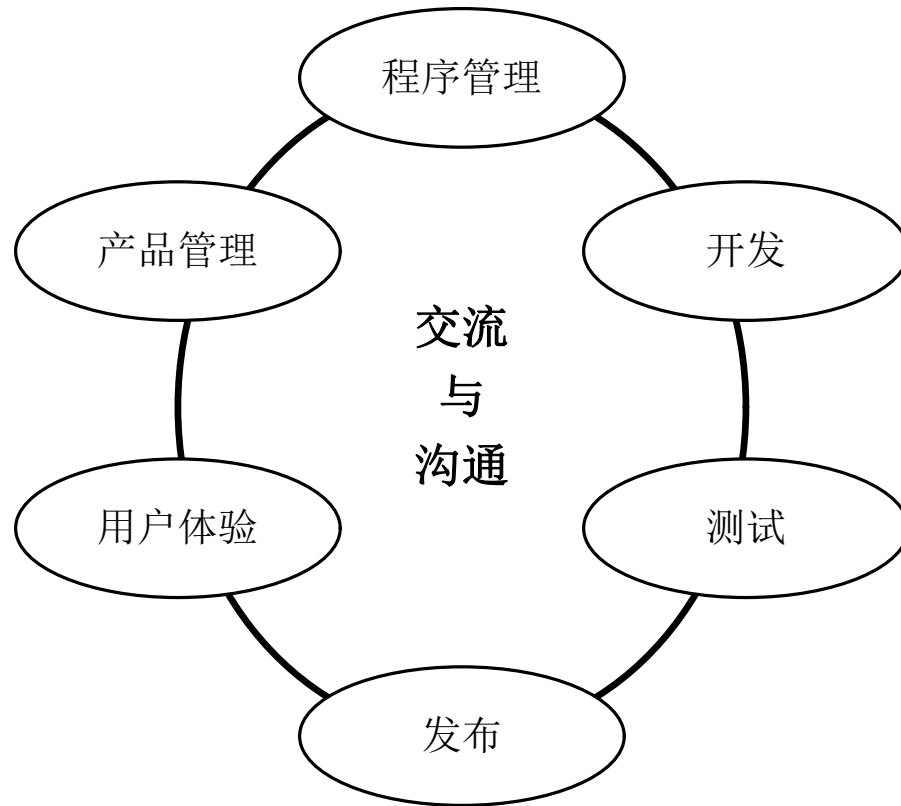
→ 为了更好地满足客户日益增长的需求，同时保证按期成功发布产品，项目组需要更有效的管理商业需求和开发技术细节的方式

2 MP的人员

- 职责设置：将传统项目经理的职责一分为二
 - 对外的用户需求管理职能交与产品经理完成，
项目组内部的综合管理职能交与程序经理完成
 - 产品经理的职责：管理用户需求
程序经理的职责：系统设计和项目组内部管理，包括编写并管理产品设计文档、组织项目组成员就开发相关问题进行交流和沟通、协调项目组日常事务
- 设置意义：以技术为基线进行管理职责划分
 - 有利于实现专家式管理
 - 保证两者的独立性和相互制约性，有利于商业需求和技术细节更好的结合

2 MP的人员

(2) 角色间的关系：对等环行项目结构



- 相互地位关系是对等
- 关键协作方式为交流与沟通

2 MP的人员

(3) 角色合并原则

- 角色合并意义

充分利用有限资源

- 多个角色可合并由一个个体担任

六种组队角色并不意味一个项目组至少需要六个人，
即一个角色不一定由一个个体担任

- 在规模较小的项目组中，某些项目组成员必须担任多个角色

2 MP的人员

■ 角色合并原则

■ 项目组内的开发人员不能兼任其它角色

- 软件开发人员是产品的创造者，他们的精力不能被分散到软件开发以外的事情中去
- 让开发人员兼任其它角色的做法无异于将产品的交付日程表交给了与产品开发无关的人员保管，项目组将无法准确控制产品开发的进度

■ 不要试图合并两个有明显利益冲突或制约关系的职能角色

- 例1：产品管理角色和程序管理角色不能合并

2 MP的人员

- 产品经理希望产品能够最大程度的满足客户需求
程序经理则希望在预定的时间和预算条件下顺利发布产品
- 如果合并
当用户想要增加需求的时候，项目组就有可能盲目接受用户的建议而不管该建议是否与当前产品中的特性或技术体系相冲突；
或者盲目改变产品特性而不管该变更是否真正反映了用户需求
- 如果保持独立
可以确保用户观点和开发观点在项目组内有同等重要的地位
- 例2：开发角色和测试角色不能合并

2 MP的人员

■ 角色合并规范表

	产品管理	程序管理	开发	测试	用户体验	发布管理
产品管理		N	N	P	P	U
程序管理	N		N	U	U	P
开发	N	N		N	N	N
测试	P	U	N		P	P
用户体验	P	U	N	P		U
发布管理	U	P	N	P	U	

其中：“N”表示不能合并，“U”表示”不建议合并，“P”表示可以合并。

■ 角色合并结论：

一个最小项目组可以只有三个成员：产品经理、程序经理和开发工程师。
其中产品经理兼任测试和用户体验角色，程序经理兼任发布管理角色

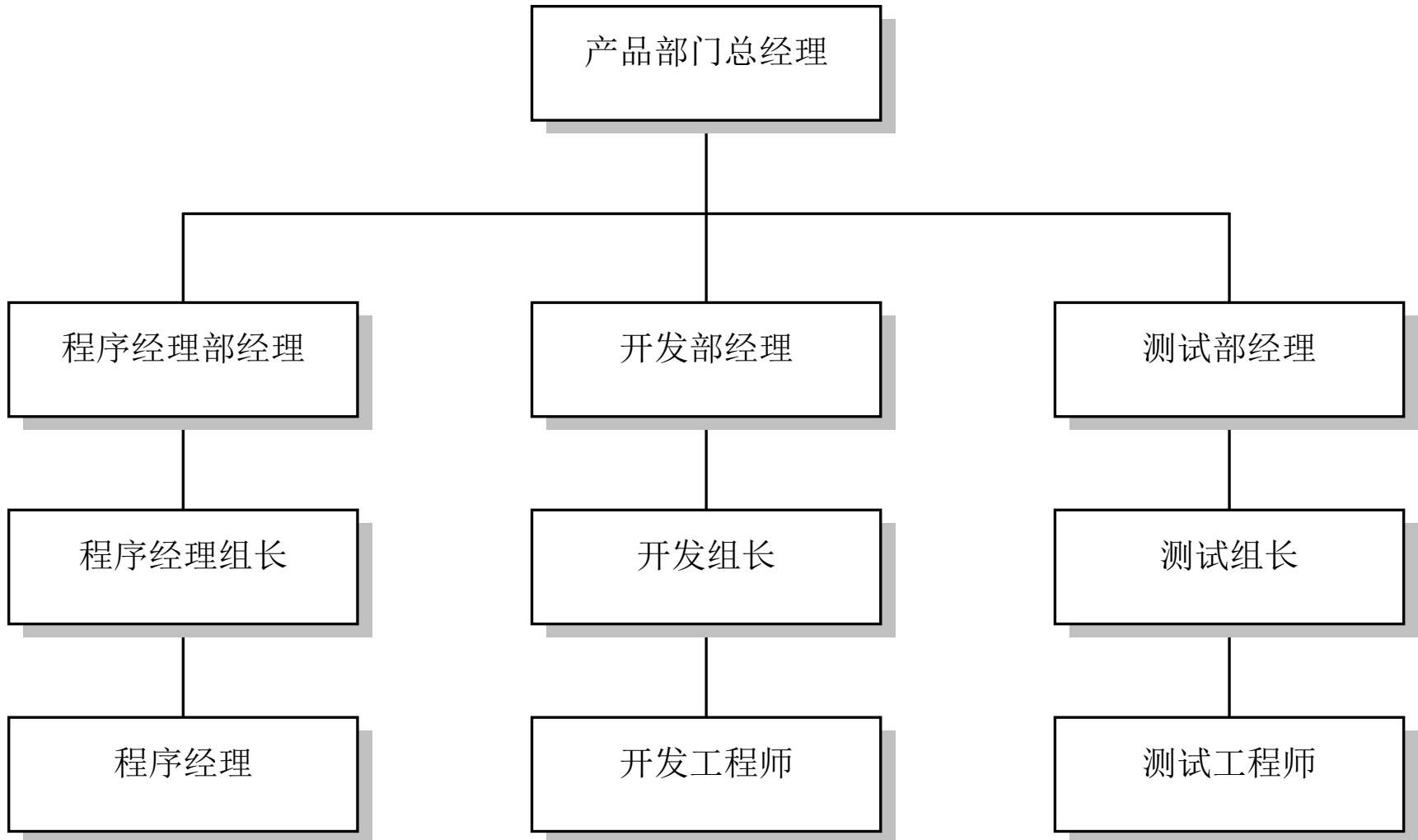
2 MP的人员

(4) 产品部门的行政组织结构

- 微软整个公司结构：七大部门
 - 四个产品部门：
 - 消费产品部
 - 平台产品部
 - 商业产品部
 - 开发工具产品部
 - 一个解决方案部
 - 一个全球销售&市场&服务部
 - 一个微软研究院

2 MP的人员

■ 各产品部门的行政组织结构：



2 MP的人员

任何一个产品部门 \cong 一个相对独立运作的小公司：

- “1+3” 的结构：

- “1” 即一个产品部门总经理
- “3” 即三个部门经理，分别是程序经理部经理、开发部经理和测试部经理，这三个部门经理平级，都直接向产品部门总经理报告

- 垂直式的专家管理模式

- 每个部门经理管理几个组长
- 每个组长管理三到五个具体工作人员

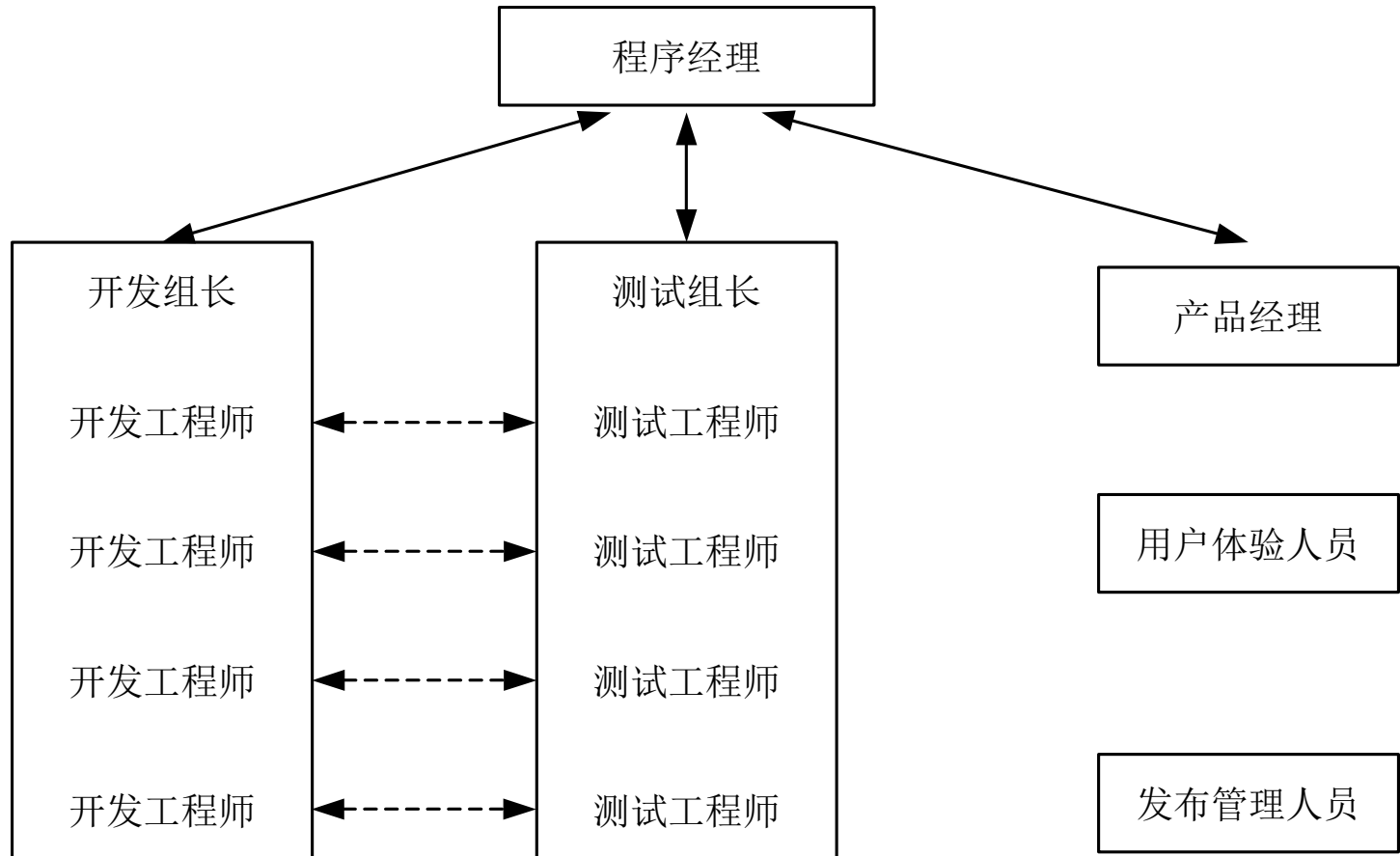
→ 构成微软的“三架马车”：对项目成功起关键性支撑作用

- 三个部门的设置对应每个项目组中最重要的三种角色——程序管理、开发和测试
 - 这三类角色是每个项目组无论怎样精简也必须的
- 其它角色来自产品部门以外的其它部门，如产品管理角色（由产品经理完成）来自市场部门

2 MP的人员

(5) 项目组的工作结构

■ 项目组的工作结构



2 MP的人员

项目组是一个对具体项目需求临时组成的工作单元

■ 人员组成：

来自产品部门的软件程序经理、开发工程师和测试工程师与来自其它部门的人员（如来自市场部门的产品经理）组成

■ 协调关系

- 程序经理负责协调各方面人员以推动整个产品的开发工作
- 图中虚线箭头表示测试组与开发组是协同工作关系（在程序经理的协调下）

➡ 人数比例

- 测试工程师和开发工程师的人数并不一定是一比一的关系
- 在微软的许多项目中，为了保证产品的质量，保证对所有产品特性进行充分有效的测试，每个开发工程师对应的测试工程师的数目不止一个。例如开发Windows操作系统时，Windows项目组中配置的测试人员数目达到了开发人员数目的两到三倍

2 MP的人员

■ 微软项目组的组建过程：

- 1) 大型项目组根据产品特性拆分为多个小型项目组，并按照小型项目组的管理原则进行管理；每个小型项目组负责产品的一个特性或一个功能模块，其人数不多于10人
- 2) 使各小型项目组并发完成工作，为此设置多个同步点，其中项目开发五个阶段的各阶段里程碑是最为重要的项目同步点
- 3) 每个项目组组长成员由六种角色构成，每种角色都有一名专家式带头人或负责人，如开发组长、测试组长，六种角色的负责人各自负责相关领域的管理、组织和协调工作；各类角色处于对等的项目工作结构中，各自完成特定的职能，他们之间主要协作方式是交流和沟通
- 4) 项目组内的角色合并（合并原则见前）

2 MP的人员

(6) 项目组人员管理的激励机制

微软的激励机制大体分为物质和精神两个层面

- 物质方面：微软被称为“高薪的血汗工厂”

- 给予开发人员很高的薪水、项目奖金、股权

- 例：微软1986年股票上市，股权认购造就了3000名百万美元的富翁

- 工作环境的创建

- 例：每栋楼的每一层物资室内常用的办公用品不需签字手续领用，大一点的办公家具只需一封电子邮件或相应位置上贴一张小纸条

2 MP的人员

- 精神方面：企业文化建设

- 每个项目小组都有一份士气基金

- 可用于做小组成员任何想做的事情，如印发小组特有的T恤衫，组织小组打保龄球、野餐、旅游等

- 为保持高涨的士气宁可以牺牲其它方面为代价

- 可能的牺牲包括软件的一致性、编程纪律、产品控制、进度计划和管理透明性等

这些机制对开发人员产生的激励作用是巨大的

例：传闻工作10小时、12小时、14小时、18小时
甚至在办公室内一连住上数周

2 MP的人员

2. 人员及组织特点

■ MP相对AP、RUP的人员特点

- RUP： 给出了人员角色种类划分，但未指出各角色间地位和工作关系
- AP： 给出了人员间相互关系和交流协作关系，但未能给人员职责的划分准则
- MP： 以上两种过程优点的结合和进一步深化发展
 - 角色划分方面： 产品管理和程序经理两权分立
 - 角色相互地位和交互关系： 相互平等，交流和沟通
 - 角色分配方面： 提出了针对有限人员限制条件下的角色合并原则
 - 项目组的规模和人员配备与管理方式上： 指出提出了由专家式行政管理 and 小型化、多元化项目组组队方式构成的矩阵结构

2 MP的人员

■ 讨论题

对比自己所在或曾经工作过的公司的人员及组织管理方式，谈谈微软公司在人员及组织管理方面采用的专家式矩阵管理模式的优点和意义。

2 MP的人员

■ MP人员组织特色及意义

- (1) 项目组由专业职责划分清晰地各对等角色组成，各角色相互配合、同时又是相互制约

- 例1：产品经理与程序经理两权分立，保证了客户观点和开发观点的独立性

- 例2：程序管理、开发、测试的三权分立式，

假设测试组归属开发组管理，则测试工作将失去独立性和质量控制的严正立场，甚至受开发人员干涉和排斥，因为开发人员大都不喜欢被指正程序失误，而更喜欢争论测试本身是否存在问题

类似，如果开发人员归程序经理直接进行行政管理，开发人员很可能为迎合领导而在开发中接受和满足程序经理在产品开发方面的干预，另一方面，开发人员也会由于程序经理不直接编程，或对自己的具体建议不被采纳而产生抵触情绪，最终影响产品开发的正常运作

2 MP的人员

■ (2) 人员的行政管理是专家式管理

- 任何一级领导都能充分了解下属的工作并为其提供专业上的指导和帮助，必要时能在下属缺席时替其完成工作任务

开发人员虚报所需工作日的情况在微软很少发生，因为开发人员清楚，自己的估算会很直接地受到自己软件开发组长的准确评判，而软件开发组长的估算在软件开发经理那里也会很容易受到检验

- 下属乐意接受能真正了解他的具体工作状况的人做领导，并给予充分的尊重

即员工不会尊敬那些没有能力却当他们所做工作的主管的人，也没有人喜欢只会管人而不懂专业的领导

➡ 这种专业合格导向的主管和明智的管理模式是微软成功的核心思维和企业文化之一

2 MP的人员

■ (3) 专业人才地培养发展遵循不同的业务具有同等的提拔机会的原则

在微软，行政级别决定员工的工资、股权等到福利待遇，而**职务头衔**与**行政级别**并不能直接挂钩的

■例1：程序员

不需要成为经理或管人的领导，只要能力强、贡献大，其行政级别及他的工资、福利待遇等可能与总经理甚至副总裁一样

■例2：测试人员

资深的测试人员可以在行政级别上得到提拔，他的能力和贡献能通过提级不断得到认可和回报

4 微软过程特点——MP与AP、RUP

- MP的生命周期

- MP的人员

- ■MP的方法

- MP的产品

- MP的相互关系

3 MP的方法

1. 关于构想阶段

- 确定项目前景和项目范围两个项目目标
 - 项目范围：是项目近期的具体目标
即第一次生命周期内的目标
 - 项目前景：是项目的远景目标
即第二次及以后若干次生命周期内的目标
- 因此，各次生命周期不同目标得以明确

3 MP的方法

■ 动态满足需求——先基线化、后冻结

■ 内容：

■ 在开发过程的前期欢迎变化

- 前期包括构想阶段、计划阶段、开发阶段
- 对主要配置项进行基线化及采用变更管理程序

■ 在开发过程的后期阶段

- 后期指稳定阶段及以后
- 对所有配置项进行冻结，一般不再允许修改

3 MP的方法

- 比较MP与AP的动态满足需求变化方法：

- 相同点

对需求变化的态度：欢迎变化

- 不同点

- AP：在第2条原则中指出，“即使到了开发的后期也欢迎改变需求”
- MP：对变化在后期进行冻结的策略应该更是现实性，因为到开发的后期还要响应变化，其成本代价和风险代价均太高，明智的策略不如将这一需求变化留至下一个版本的开发中实现

3 MP的方法

2. 计划阶段

- 以产品特性及其优先级指导整个项目
 - 提高了产品的竞争力
 - 降低了产品开发各阶段的风险

3 MP的方法

3. 关于开发阶段

■ 代码优化

代码的性能包括运行效率、安全性、稳定性、可理解性、可维护性等多个方面，例：

■ 代码的运行效率

- 优劣算法之间的差距靠硬件的投入平衡不了
- 根本上，提高代码运行效率必须不断优化算法、数据结构和程序结构

3 MP的方法

■ 高信度计算 (Trustworthy Computing)

■ 提出的目的

- 微软公司针对目前IT领域内存在的黑客攻击、网络入侵、病毒传播和软件漏洞等一系列影响IT系统运行、阻碍正常信息交流的安全问题提出
- 旨在提高软件、应用系统和网络的安全性，保证信息安全部署

■ 内容措施

- 发布前确保系统的默认设置是安全的
- 防范缓冲区溢出攻击
- 永远不要相信用户输入的数据内容等

3 MP的方法

- 代码优化方法

- 参见

- 林锐的“高质量C / C++编程指南”

- 课后结合自己曾经编写的软件进行优化

- 写一篇代码优化方法总结

- 总结内容包括优化前的源代码、优化后的源代码、优化原理分析

3 MP的方法

- 源代码管理

建立源代码的管理库，每日Check-in进行持续更新和集成

- 每日编译生成

类同AP的持续集成（详见前述部分）

3 MP的方法

关于稳定阶段

- 零缺陷管理
(详见前述部分)

3 MP的方法

■ 手工测试与自动测试结合

■ 内容

- 手工测试是依靠人力查找Bug
- 自动测试是通过编写一些测试工具，自动运行查找Bug

■ 优缺点

■ 自动测试

- 优点：查找速度快、范围广
- 缺点是：只能检查一些最重要的问题，如崩溃、死机，但都无法发现一些一般的日常错误；且在自动测试中编写测试工具的工作量也很大

- 手工测试：人眼容易找到一些一般的日常错误

➡ 实际测试中：通常手工测试与自动测试相结合

3 MP的方法

■内部测试与外部测试结合

■Microsoft公司经历的惨痛教训

80年代初期，Microsoft公司的许多软件产品出现了Bug。比如，在1981年与IBM PC机一起推出的BASIC软件，用户在用“.1”（或者其他数字）除以10时，就会出错；在FORTRAN软件中也存在破坏数据的Bug。由此激起了许多采用Microsoft操作系统的PC厂商的极大不满，而且很多个人用户也纷纷投诉。Microsoft公司的经理们发觉很有必要引进更好的内部测试与质量控制方法，但是遭到很多程序设计师甚至一些高级经理的坚决反对，他们固执地认为在高校学生、秘书或者外界合作人士的协助下，开发人员可以自己测试产品。在1984年推出Mac机的Multiplan（电子表格软件）之前，Microsoft曾特地请Arthur Anderson咨询公司进行测试，但是外界公司一般没有能力执行全面的软件测试。结果，一种相当厉害的破坏数据的Bug迫使Microsoft公司为它的2万多名用户免费提供更新版本，代价是每个版本10美元，一共化了20万美元，可谓损失惨重。痛定思痛后，Microsoft公司的经理们得出一个结论：如果再不成立独立的测试部门，软件产品就不可能达到更高的质量标准，IBM和其它有着成功的软件开发历史的公司便是效法的榜样。

3 MP的方法

但Microsoft公司并不照搬IBM的经验，而是有选择地采用了一些看起来比较先进的方法，如独立的测试小组等。Microsoft公司的一位开发部门主管戴夫·穆尔回忆说：“我们清楚不能再让开发部门自己测试了。我们需要有一个单独的小组来设计测试，运行测试，并把测试信息反馈给开发部门。这是一个伟大的转折点。”但是有了独立的测试小组后，并不等于万事大吉了。自从Microsoft公司在1984年与1986年之间扩大了测试小组后，开发人员开始“变懒”了。他们把代码扔在一边等着测试，忘了唯有开发人员自己才能阻止错误的发生、防患于未来。此时，Microsoft公司历史上第二次大灾难降临了。原定于1986年7月发行的Mac机的Word 3.0，千呼万唤方于1987年2月问世。这套软件竟然有700多处错误，有的错误可以破坏数据甚至摧毁程序，一下子就使Microsoft名声扫地。公司不得不为用户免费提供升级版本，费用超过了100万美元

3 MP的方法

■ 教训总结

- 内部测试：需要开发人员与独立的测试小组共同参与
 - 开发人员执行“白盒”测试，即根据源程序内部逻辑结果测试程序的过程性细节
 - 而独立测试小组执行“黑盒”测试，即完全不考虑程序内部的逻辑结构，只依据程序的需求规格说明书检查程序的功能是否符合其功能说明
 - 对于没有条件设立独立测试小组的公司，可采用同级评审测试的方法，即让开发小组的成员相互测试对方的程序
- 外部测试：

交于专门的、权威的测试机构以及现场用户进行测试

3 MP的方法

5. MP的工具

微软的第一个涉及软件开发全生命周期的产品—Visual Studio Team System, 包括:

- Visual Studio 2005 Team Architect Edition
 - 提供基于微软的领域规范设计语言（DSL）构建的四部分建模方法
 - UML2.0视为DSL中的一种，微软认为UML不够精确，难以映射到诸如ASP.NET等低层框架

3 MP的方法

- Visual Studio 2005 Team Developer Edition

静态分析、代码剖析、代码涵盖、单元测试工具

- Visual Studio 2005 Team Test Edition
管理和运行测试（包括单元测试、手工测试和Web 测试）的工具

- Visual Studio 2005 Team Foundation

- 项目管理工具，包括 Microsoft Project、Windows Share Point Services等

- 源代码管理工具VSS（Visual SourceSafe）等

4 微软过程特点——MP与AP、RUP

- MP的生命周期
- MP的人员
- MP的方法
- ■MP的产品
- MP的相互关系

4 MP的产品

■ 产品类型

- 各类文档：如前景/范围说明书、功能说明书、项目计划、总结报告等
- 源代码
- 可执行文件
- 相应的文档代码库

■ 各类产品的优先级

未进行专门阐述

■ 产品的功能与性能

以产品特性及优先级指导整个项目

4 微软过程特点——MP与AP、RUP

- MP的生命周期

- MP的人员

- MP的方法

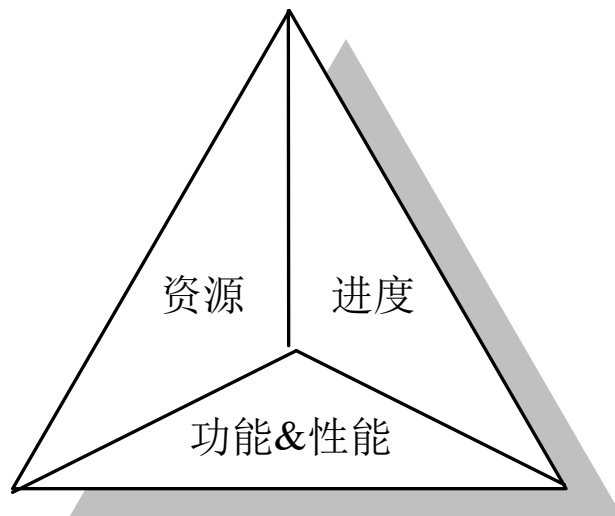
- MP的产品

-  ■MP的相互关系

5 MP的生命周期、人员、方法、产品四要素间的相互关系

■ 均衡三角形

- 过程的生命周期进度、人员及方法工具等项目资源、产品的功能和性能之间存在一种相互制约的均衡三角形关系
- 任何对三角形一边的改变都将导致三角形另外一边或两边的变化，因为只有这样才能保持三角形关系的均衡
- 发布一个符合客户需求的产品，其关键在于项目组必须在进度、资源、产品功能&性能之间的三角型寻求最佳的平衡点



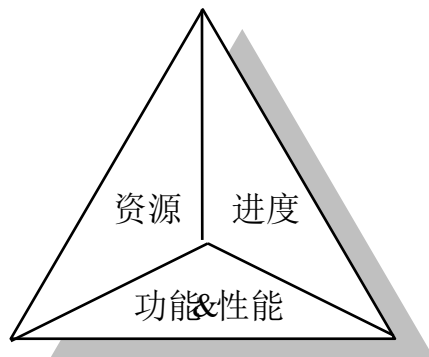
5 MP的生命周期、人员、方法、产品四要素间的相互关系

■ 项目均衡矩阵：有效的项目均衡三角形管理工具 定义了资源、进度和功能&性能三者需要做出折中时的优先级关系

■ “能接受”表示：
不可改变的或是
最低要求的

■ “最佳”表示：
必须优先考虑、
作出最好选择的

■ “可调整”表示：
既不是最佳也不是
仅能接受，必须根
据情况做出相应的调整，以保持三角形的均衡



可调整

最佳

能接受

资源

进度

功能&性能

✓		
	✓	
		✓

5 MP的生命周期、人员、方法、产品四要素间的相互关系

■ 理解项目均衡矩阵：

将“资源”、“进度”和“产品功能&性能”这三个词填充到下面这句话的空白处：

“在_____一定的情况下，我们可以选择_____，并对_____做必要的调整”

■ 逻辑组合结果：可作为项目组和客户在权衡、调整项目各要素时的指导

- 在资源一定的情况下，我们可以选择进度，并对产品功能&性能作必要的调整
- 在资源一定的情况下，我们可以选择产品功能&性能，并对进度作必要的调整
- 在产品功能&性能一定的情况下，我们可以选择资源，并对进度作必要的调整
- 在产品功能&性能一定的情况下，我们可以选择进度，并资源对作必要的调整
- 在进度一定的情况下，我们可以选择资源，并对产品功能&性能作必要的调整
- 在进度一定的情况下，我们可以选择产品功能&性能，并对资源作必要的调整

5 MP的生命周期、人员、方法、产品四要素间的相互关系

■ MP四要素相互关系的特点

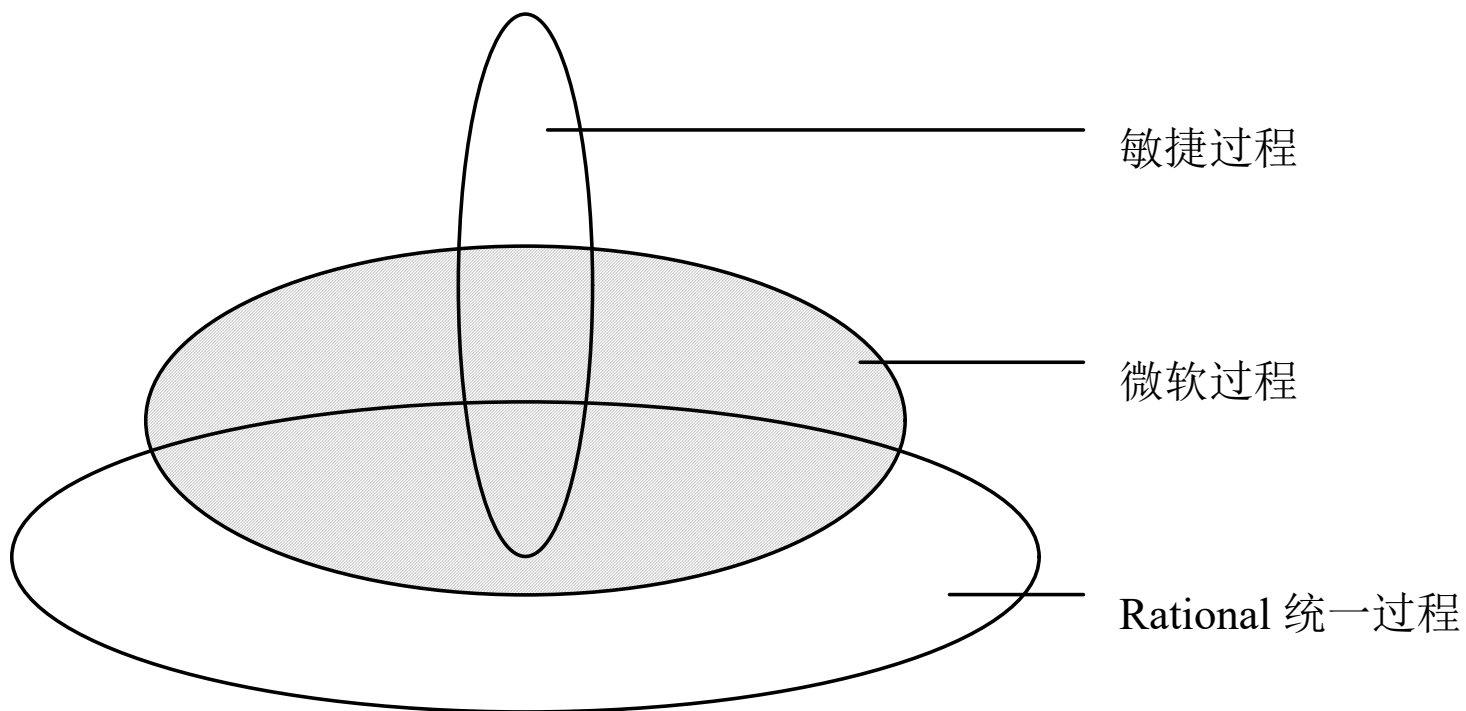
相对AP提出的“个体交互胜过过程和工具”这一绝对的观点，MP四要素之间的均衡关系总结

- 更全面、更具普适性
- 不同的项目可根据项目的具体环境需要对这四要素做出合适的优先级安排、寻求各要素之间的最佳平衡点

4 微软过程特点——MP与AP、RUP

■小结：MP与AP、RUP三者之间关系

见下图，三者相互交叉重叠又相互区别、互不包含



4 微软过程特点——MP与AP、RUP

其中：

- RUP：一个大而全的过程框架，可适应于各种具有不同项目环境类型的项目开发，包括理想的项目开发环境和具有有限资源与时间约束的项目环境
- AP：针对商业环境中具有有限资源和时间进度限制的小型项目提出的一种软件过程模式，对RUP的进行了多方面的有益补充和完善
- MP：另一种针对商业环境中具有有限资源和时间限制项目的软件过程模式，综合了RUP与敏捷过程的诸多优点，并且进一步在人员及其组织、过程中的方法等方面提出了独具特色的、操作性很强的实践规范，是一套优秀的成功项目开发实践经验总结

5 微软过程实施策略

■ 实施策略：

- MP适用范围：有限资源和有限时间约束的项目
- 运用微软过程中各项经验原则时，应良莠区分，不可全盘照抄
 - 总体上微软过程是一套优秀的软件过程模式，它综合了RUP和敏捷过程的诸多优点，并且在很多方面对前面两者进行了改进和补充完善
 - 也存在一些缺陷：如在进度计划的制定方面，MP提出制定确定的计划不如AP提出的制定细致度逐渐降低的计划所具有的对变化的适应能力和灵活性
- 缺项内容可参考借鉴RUP和敏捷过程的相应原则规范
 - 微软过程对方法工具、产品等因素的论述不及RUP和敏捷过程全面
- 根据具体项目的要求相应进行裁剪、扩充等配置操作
 - 因为项目是多样性的，不同项目具有不同的项目开发环境

6 微软过程案例——IE浏览器经典战役

Internet Explore（IE）浏览器案例：

浏览器市场竞争的经典成功案例

- 具体分析阐述IE产品开发过程中的软件过程模式四要素内容及相互关系
- 同时说明一些具体操作上的细节问题

6.1 浏览器市场之争案例描述

1995年到1996年间，Netscape公司的Navigator浏览器占据了全球90%以上的市场份额。当时，大多数人上网浏览、冲浪的首选工具都是Netscape的Navigator，很少见到有谁在使用微软公司IE浏览器。

1995年12月7日，比尔·盖茨在公司大会上宣布：“今天是微软公司的Internet日”。这表明，从那一天开始，微软公司所有员工都必须认真分析和思考Internet对软件企业的挑战，并全力以赴在微软全系列软件产品中增加对Internet支持。1996年初，微软成立了IPTD产品部门，该部门主要负责开发IE浏览器和Java语言工具这两类产品，部门汇集了许多来自Windows95等产品部门的资深工程师。当Netscape公司听说微软成立了IPTD部门时，他们丝毫不为所动。Netscape公司早在1995年就曾经针对微软推出的IE1.0版本向外界宣称：“我们领先微软二年，他们永远也别想追上我们”。到1996年，微软发布了IE2.0版本，Netscape有说话有了一些改变：“我们领先微软一年，他们永远也别想追上我们。”

6.1 浏览器市场之争案例描述 (续)

1996年末，IE3.0版问世。由于IE3.0功能丰富、性能卓越，特别是IE3.0基于组件的核心技术非常适用于定制和开发，美国最大的Internet服务商——美国在线（AOL）最终决定使用IE3.0作为其预设的浏览器软件。

1997年，新推出的IE0在几乎所有媒体评测中都击败了Netscape的产品。

1998年11月，美国在线（AOL）宣布收购Netscape公司。

1999年，IE5.0向全球用户展示了卓越的运行效率和稳定性，这时，市场上已经没有人还在对Netscape念念不忘了。

2001年，IE6.0面世。作为一种优秀的产品，IE占据了93%~94%的市场份额，而Netscape的市场份额则下降到了3.5%，微软开发IE6.0时假想竞争对手实现上就只剩下微软公司自己的IE5.0了。浏览器市场之争到此已经硝烟散尽。

6.2 IE项目的生命周期

1. IE的各次版本

发布时间	版本	开发人员数目 (约)
1994 年冬季	IE1.0	7
1995 年11 月	IE2.0	30
1996 年8 月	IE3.0	70
1997 年6 月	IE4.0	300
1999 年8 月	IE5.0	500
2000 年7 月	IE5.5	300
2001 年8 月	IE6.0	100
2002 年10 月	IE6.0QFE (快速漏洞修补)	5

6.2 IE项目的生命周期

- IE3.0、IE0两个版本:开发均用了仅十个月的时间
 - 与Navigator的激励迫使微软必须在尽可能的时间内不断推出新的版本来争取更多的客户，从而逐步动摇Navigator在市场的霸主地位
 - IE3.0、IE0能够在发布时间上抢先Netscape的原因：
 - 项目组的努力工作
 - 项目组所有成员几乎每天都从早上9点工作到晚上12点，每周工作7天，没有周末
 - 全公司的全力配合
 - 整个公司都非常关注IE项目组的工作，主管副总裁每天都会打电话或发电子邮件询问项目进展情况
- IE5:开发时间为两年
 - 当IE0基本站稳脚跟后，微软用了两年多的时间精心打造出具有卓越性能的IE5.0
- 以后每个版本：开发时间为一年多的时间

6.2 IE项目的生命周期

2. IE每个版本的生命周期

例：IE5.0版本生命周期的各个阶段及里程碑设置

所属阶段	日期	里程碑
构想阶段	6月25日	前景与范围说明
计划阶段	7月1日	单页功能说明书
	7月15日	详细功能说明书
开发阶段	9月1日	平台代码（指浏览器引擎部分的代码）开发完成
	10月8日	用户界面代码开发完成
	11月7日	发布候选版本。候选版本意味着项目组认为产品已经非常接近可发布状态，并可以开始最后的测试了

6.2 IE项目的生命周期

稳定阶段	11月9日	发布 Beta-1 版本。这一版本主要供内部测试使用，即让每一个内部员工参与测试。微软的很多员工都配备两台计算机，一台用于日常工作，另一台就可以用于公司产品的测试。内部员工在测试过程中通常能发现许多产品缺陷或错误并能为项目组提出大量有价值的建议。
	4月5日	发布 Beta-2 版本，进行公开测试。公开测试主要指公司外部用户的测试，对 IE 产品来说，所有用户都可以从 Internet 上下载 IE 浏览器的 Beta-2 版本，并安装到自己的计算机上。微软也把产品的公开测试版本提供给他的小伙伴，如 IBM、HP、DELL 等企业进行测试。
发布阶段	7月12日	发布正式版本（RTM）。这意味着 IE 本次版本生命周期的成功结束。

6.3 IE项目的人员及组织

1. 人员角色配备

■ 例：IE0—共有约300名工作人员

□ IE 产品部门总经理：	1 人
□ 产品经理：	20 人
□ 程序经理：	50 人
□ 开发工程师：	100 人
□ 测试工程师：	100 人
□ 用户体验人员：	10 人
□ 发布管理人员：	5 人

6.3 IE项目的人员及组织

■ 点评:

■ 测试人员的配置方式

- 开发工程师有100人，测试工程师也有约100人，即项目组中每个开发工程师都配有一个结伴的测试工程师
- 这种测试配置有利于项目组对产品的每一个细节进行周密、完善的测试

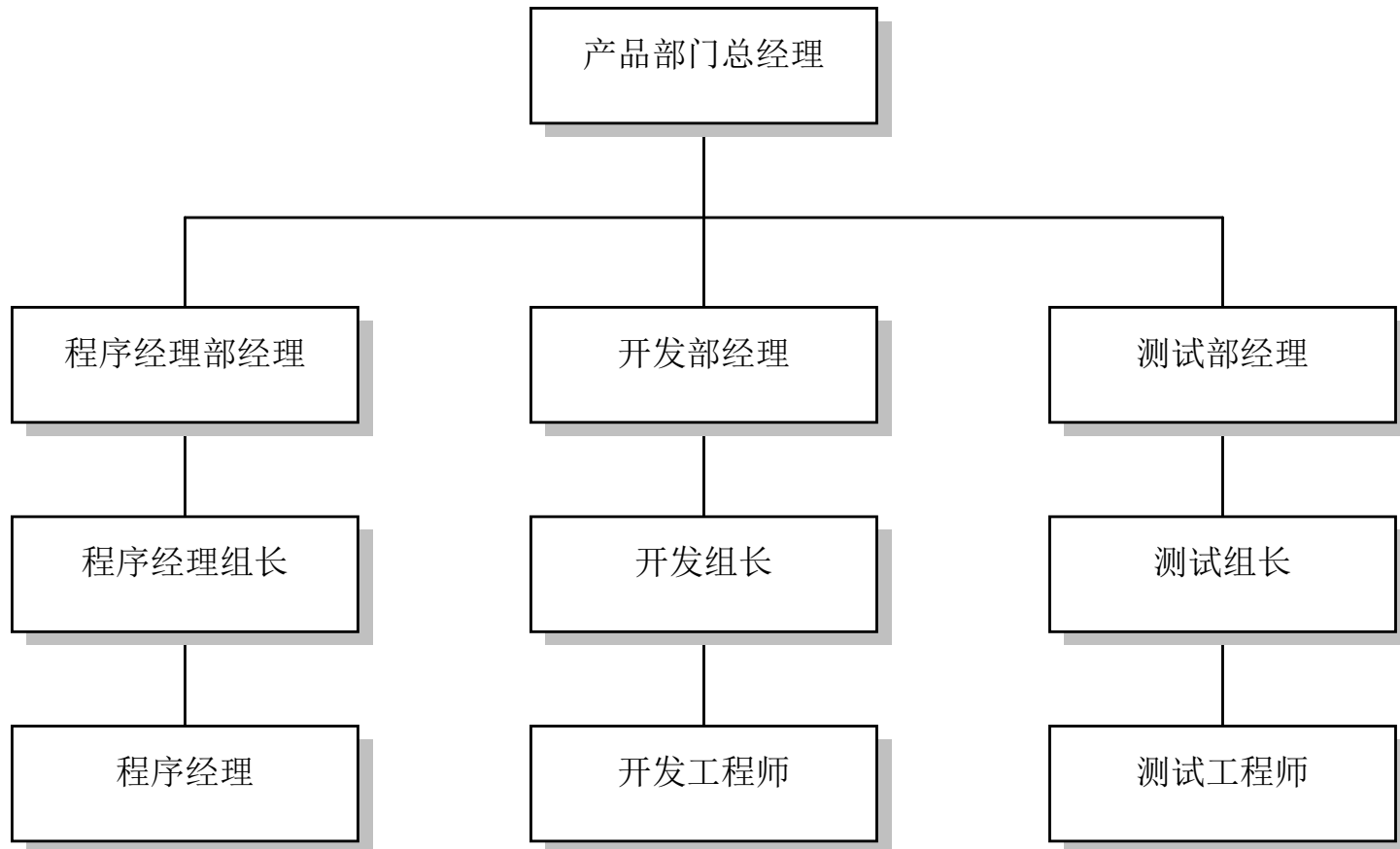
■ 微软产品拥有高质量文档的原因

专人负责设计文档和用户文档的编撰

- 50位程序经理负责编写产品的功能说明书等设计文档、管理项目进度、推动项目进展
- 10名用户体验人员专门负责编写用户手册、联机帮助等用户支持文档

6.3 IE项目的人员及组织

2. 人员行政组织结构



6.3 IE项目的人员及组织

3. 项目组的组建及工作结构

■ IE产品特性划分：三大部分

■ 用户界面部分

包括菜单、按钮、窗口和对话框等用户界面元素及其相关操作

■ 浏览器引擎部分

■ 负责调入HTML页面并显示HTML的内容的核心组件

■ 服务器端应用部分

■ 当时IIS服务及相关应用也由IE产品部门负责开发

6.3 IE项目的人员及组织

■ IE产品部门的划分：

- 根据三大产品特性划分为三个大的项目组
- 用户界面和浏览器引擎这两个大的项目组又划分成规模更小的产品特性项目组
- 每个产品特性项目组大约有5~10名成员组成，各成员角色包括1个产品特性项目组组长和若干程序经理、开发工程师、测试工程师、用户体验人员和发布管理人员
- 不同产品特性项目组在产品部门的统一协调下并行地完成IE0的开发工作

6.4 IE项目的方法

1. 构想阶段

- 构想阶段成功的市场需求分析是关键因素之一
 - (1) 关注关键需求
 - 曾收集到来自不同方面的大量不同需求信息
 - 如果实现所有可以实现的功能，产品就会异常庞大，项目开发进度也会一拖再拖
 - 最终决定：只关注那些用户最常使用的、对市场竞争起关键作用的需求
 - (2) 仔细研究竞争对手
 - 关注Netscape公司的运营情况
 - Navigator版本中有哪些用户喜欢的新功能

6.4 IE项目的方法

- (3) 关注Internet新用户，不去花费精力争取Netscape的老客户
 - 为那些刚刚接触Internet、没有太多上网浏览经验的新用户特别设计了许多简单易用的功能
 - 赢得了大多数Internet新用户的支持, 市场份额大幅度上升

6.4 IE项目的方法

■ 基于以上市场需求分析，最终确定

■ IE0项目前景

“此次计划推出的IE浏览器将成为Internet上的主流浏览器软件，其市场目标是在1998年将IE的市场占有率扩大到65%”

■ IE0项目范围

“该IE浏览器可以为企企业客户和最终用户提供高速、稳定、总体拥有成本最低的使用体验，可以与微软的Office有效集成”

6.4 IE项目的方法

2. 计划阶段

■ 编写单页功能说明书

■ 编写要求

- 由程序经理编写，一个程序经理一般同时负责3~5个产品特性
- 每一个产品特性一份一页纸的功能说明书

例：IE的某一个版本中，需要开发100个产品特性，则程序经理需要编写100份单页功能说明书，对于50个程序经理，每个程序经理负责2个产品特性

■ 编写目的

- 让开发工程师、测试工程师和其他项目组成员清楚每项产品特性的功能、操作方式和接口等信息，了解为什么需要这些产品特性（客户的直接需要还是市场竞争的需要）
- 依据市场需要的紧迫程度和该特性在产品构架中的地位等因素确定各产品特性的优先级，以此安排具体的开发任务顺序

6.4 IE项目的方法

■ 编写详细功能说明书：

■ 详细功能说明书基本章节结构

- 责任人/作者 (Owners)

- 概述 (Summary)

- 指导原则 (Guiding principles)

- 情境设计描述 (Scenario design)

- 站在用户的角度思考，想象用户在使用产品某一个功能时具体的操作方式可能是什么样的，用文字将想到最好方案描述出来，即所谓的～
- 通常是依据可用性实验室的研究结果做出
- 有助于更好地理解用户需求、设计更好的产品操作流程和用户界面

6.4 IE项目的方法

- 产品特性设计 (design)
- 安全设计 (Security)
 - 微软面临的安全问题比较多，网络上的黑客攻击和病毒入侵事件层出不穷
 - 尽量在设计阶段就做到防微杜渐、未雨绸缪，确保产品具有足够的安全性
- 安装和发布 (Setup and deployment)
- 国际化、本地化 (Globalization/Localization)
 - IE是一种国际化的软件产品，支持28种以上语言，因此设计是必须考虑到目前的技术实现是否便于向其它语言环境的移植
 - 设计过程中也必须考虑一些国家特殊的风俗习惯等社会因素
- 存在问题 (Open issue)
 - 在设计阶段还没有明确答案的问题
- 更新记录 (Change history)

6.4 IE项目的方法

■ 详细功能说明书的变更管理

作为一份活动的设计文档

- 每一次对设计文档的修改都必须遵循严格的变更管理程序
 - 在变更历史中留下清晰的更新记录
- 以便项目组在任何时候都可以比较容易地跟踪文档的变更过程

6.4 IE项目的方法

3. 开发和稳定阶段

■ 源代码及文档管理

- IE项目组使用一个统一的数据库管理源代码、文档等资源
- 每天开发结束后，开发工程师将自己计算机上的最新源代码和文档更新到数据库中，这个过程被称为检入（Check-in）过程，包括
 - 同步代码，每个人先生成自己版本。保证新代码与原版本不发生冲突
 - 在检入前做代码审查。由另一名有经验的程序员帮助开发者阅读和审查代码
 - 检入时，代码必须满足检入条件。检入条件包括通过BAT、BVT和其它测试
 - 遵守检入窗口制度。大型项目组一般设置检入窗口，不同产品特性开发组人员在不同的规定时间检入他们的代码，这样，如果产品生成失败，容易查找Bug的位置
 - 发送检入邮件通知项目组
- 所有开发工程师都检入自己最新的代码后，项目组就可以编译整个程序，完成每日产品生成的工作

6.4 IE项目的方法

■ 每日编译生成

每天都编译并生成完整的可执行版本，这是微软公司最为重要的开发制度之一。其内容包括：

- 整个生成过程都是自动完成的
- 每天的同一时间，通过同步所有项目组件，创建一个源代码树的拷贝
- 编译生成所有的组件
- 运行BAT和BVT测试，检验生成版本的可用性
- 向项目组发送状态报告邮件
- 发送每日同步日志，在公共服务器上公布生成的产品版本

6.4 IE项目的方法

■ Bug数据库管理

每个产品都有一个集中的Bug数据库，其管理工作主要包括：

- 由测试工程师创建Bug记录。产品中的Bug一般包括代码缺陷和不完善的产品特性等
- 程序经理负责每天审查Bug数据库，并为开发工程师分配Bug修改工作
- 开发工程师修正Bug，并将结果发回给测试工程师
- 测试工程师使用每日生成来检验Bug是否已经修正，并修改Bug记录。若确定已修正，则关闭该项Bug

6.4 IE项目的方法

发布阶段

- 使用各种发布渠道向用户免费发放IE浏览器
 - 微软公司不向客户收取使用IE浏览器的任何费用
 - IE浏览器不仅对个人用户是免费的，对所有企业用户也是免费的
 - 在微软公司的Windows操作系统中捆绑IE浏览器软件

尽管这一做法在后来引发了著名的微软公司垄断案，但它的确是微软公司在浏览器市场取得成功的重要因素之一

6.4 IE项目的方法

■客户群细分

- 不仅对个人用户提供足够的支持
- 也对企业用户、包括大型企业用户提供完善的服务

→ 这种全面、细致的营销和服务体系帮助微软赢得了大多数客户的信任和支持

6.5 IE项目的产品

包括

- 各版本的源代码
- 可执行文件
- 项目文档，如
 - 项目前景/项目范围
 - 单页/详细功能说明书等

6.6 IE项目的生命周期、人员、方法与产品四要素间的关系

IE项目中采用的四要素的均衡矩阵为：

在进度（即过程的进度时间）一定的情况下，我们可以选择产品的功能&性能，并对资源（包括人员、方法等）作必要的调整。

其中：

- “进度一定”是指微软在每个IE版本发布新功能时都应总是比Netscape抢先一步
- “可以选择的产品功能&性能”是指为每个版本选择关键特性，并且每个版本之间持续、递进的交付
- “对资源作必要的调整”是指，如人员的配备在IE1.0时为7人，到IE5.0时猛增至500人

7 本章小结

- 微软解决方案框架（MSF）及其与微软过程的关系
- MP规范内容：过程原则+组队原则
- MP相对AP、RUP的特点：从软件过程模式角度
 - 生命周期：五个阶段
 - 人员：对等环行结构，专家式的矩阵管理模式
 - 方法：各阶段一系列操作性很强的方法
 - 产品：项目范围/前景，单页/详细功能说明书
 - 四要素关系：均衡三角形，均衡管理矩阵

7 本章小结

■ MP的优缺点

- 作为另一种针对商业环境下具有有限资源和有限时间限制的项目的软件过程模式，MP综合了RUP和AP的诸多优点，并且进一步在人员及其组织、过程中的方法等方面提出了独具特色的、操作性很强的实践规范，是一套优秀的成功项目开发实践经验总结
- MP也存在某些缺陷，如对方法工具、产品等方面的论述不及RUP和AP全面，有些原则本身也存在问题

■ MP实施策略

结合RUP和敏捷过程一起实施

针对不同的项目进行具体的定制

自测题

- 1. 微软过程中程序经理是不是就是传统的项目经理？两者之间有何区别？
- 2. 相对RUP和AP，微软过程的生命周期有何特点？
- 3. 微软项目组是如何组建的？
- 微软过程中最具特色的是其矩阵式的人员与组织管理结构，阐述该矩阵结构的特点与意义。

自测题

- 5. 微软过程中的动态满足需求的方法与敏捷过程中动态满足需求的方法有何异同点？
- 6. 什么是微软过程中的项目均衡三角形和项目均衡矩阵？它们有何作用？
- 7. 微软过程这一软件开发过程的适用范围是什么？
- 8. 说明MP与RUP、AP三者之间的关系。