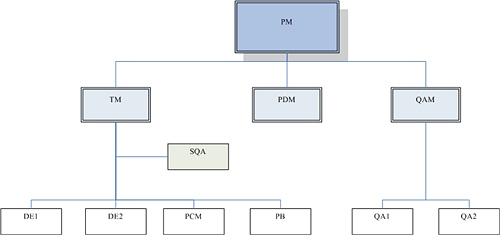
# 组织结构

# 国内

* **标准岗位列表**

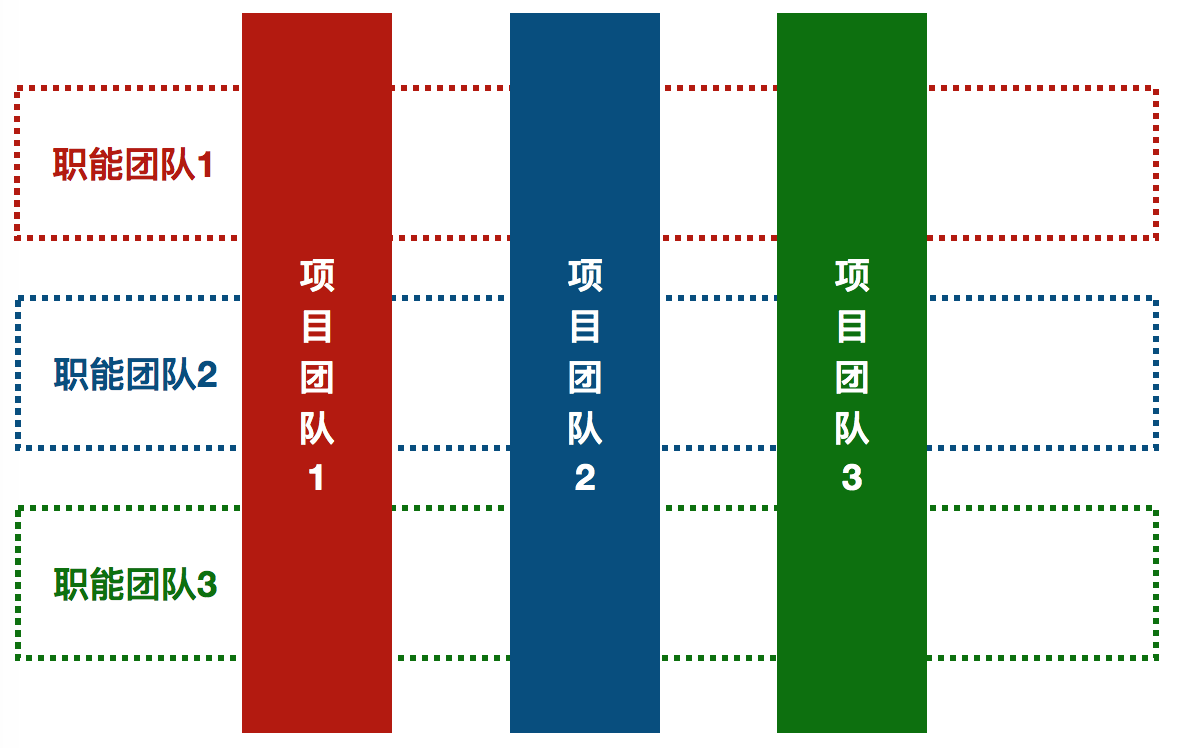
|  |  |  |  |
| --- | --- | --- | --- |
| **角色名** | **简称** | **职责** | **是否必选** |
| 项目经理 | PM | 负责整个项目的计划、执行、跟踪、发布。  是项目的最终决策人和负责人 | Y |
| 技术经理 | TM | 负责项目的代码开发管理，包括：预估开发工作量、开发任务分配、  开发日报发布、详细设计、代码BUG跟踪、发布前准备。  是在技术开发方面的最终负责人。 | Y |
| 产品设计经理 | PDM | 负责项目的功能性需求与产品设计。  是产品功能设计与交互设计的最终负责人。  也是项目DEMO演示时的主要参与者之一。 | Y |
| 运营经理 | POM | 负责项目的非功能性需求，主要是运营、推广方面的设计与实施。  也是项目DEMO演示时的主要参与者之一。 |  |
| 测试经理 | QAM | 负责项目的测试计划、测试任务分配、BUG跟踪。  是项目最终是否通过测试，是否可以上线的最终决策人。  也是项目DEMO演示时的主要参与者之一。 | Y |
| 客服经理 | CSM | 是项目的来自于客户需求的管理者。  也是项目DEMO演示时的主要参与者之一。 |  |
| 开发组长 | TL | 负责某一个开发小组的管理任务。受TM领导。 |  |
| 架构师 | PA | 负责整体架构，代码审核。 |  |
| 开发工程师 | DE | 负责具体的设计、编码、测试。受TL或TM领导。 | Y |
| 数据库管理员 | DBA | 负责项目的数据库设计的审核、变更，SQL语句的审核，数据订正，测试或上线数据准备。 |  |
| 系统管理员 | SA | 负责环境部署、维护；网站流量或压力监控。 |  |
| 界面设计师 | UI | 负责用户界面设计与交互设计。 |  |
| 产品运营师 | PO | 负责产品的运营活动 |  |
| 产品设计师 | PD | 负责需求的收集与分析，产品设计与交互设计。 |  |
| 测试工程师 | QA | 负责软件的集成测试，包括三个环境下的测试：测试环境、予发布环境、线上环境。 | Y |
| 配置管理员 | PCM | 负责本项目的分支管理，代码合并，版本控制。 | Y |
| 发布员 | PB | 负责发布程序到：开发环境，测试环境，线上环境。 | Y |
| 软件过程专员 | SQA | 负责监督项目的过程计划与执行情况。协助项目的规范化以及风险告警。  目前SQA专员仅协助TM，关注于开发过程的管理与执行。 | Y |

* **图示**



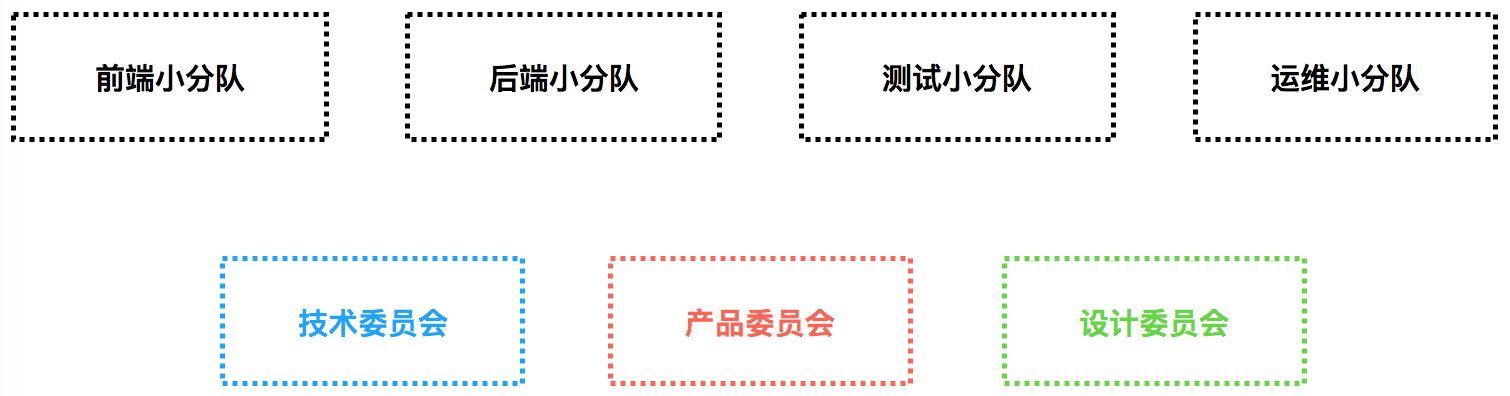
## 矩阵式组织架构

如果研发团队规模大于 10 人，并且希望团队以最高效的方式实现项目交付



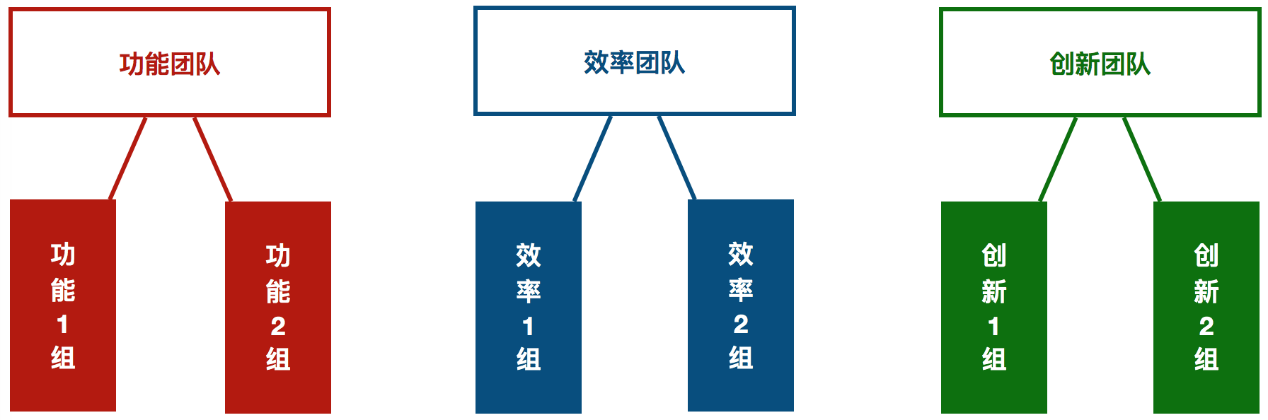
## 横向的职能团队

根据团队成员专业技能的不同，可划分为多个职能团队，也称为“小分队”，例如：前端小分队、后端小分队、测试小分队、运维小分队等。当然，可根据我们所面临的实际环境，灵活划分出合理的职能团队。



## 纵向的项目团队

在纵向层面，我们还需要搭建一些项目团队，并确保这些项目团队是可以并行工作的，也就是说，他们的工作一般是彼此隔离，不会相互干扰。



# 国外

* **项目的类型和复杂性**

|  |  |  |
| --- | --- | --- |
| **项目范围/阶段** | 团队规模 | 软件开发团队角色 |
| 发现/概念证明 | 最多5名专家 | 产品负责人（通常在客户端）、项目经理、业务分析师、软件架构师、UI/UX 设计师 |
| MVP 开发 | 6+专家 | 产品负责人（通常在客户端）、项目经理、业务分析师、UI/UX 设计师、软件工程师、测试工程师 |
| **产品开发** | 敏捷：通常最多 9 名专家；如果一个项目是大型的，几个敏捷团队可能会一起工作。瀑布：没有团队规模限制；具体人数将取决于申请的类型和复杂性 | 产品负责人（通常在客户端）、项目经理、业务分析师、UI/UX 设计师、软件架构师、软件工程师、测试工程师 可选：测试自动化工程师、性能工程师、DevOps 工程师、安全工程师 |

## 软件开发团队的成员

## 产品拥有者

* 负责产品愿景和发展
* 确保最终产品符合客户要求

与任何其他开发团队成员相比，产品所有者对产品的成功负有更多责任，他是决策者。他们平衡业务需求和市场趋势，定义业务战略，塑造产品愿景，确保满足客户需求，并管理产品积压。主要与灵活的敏捷环境相关联，产品负责人必须在需求和工作流程经常变化的场景中很好地工作。BA 和 PO 的职责听起来非常相似。两者有什么区别，在一个项目中是否需要两者？关键的区别在于，产品负责人提供产品的愿景，而不深入研究其技术实现方式，而业务分析师则在客户和团队之间架起桥梁，更多地关注技术方面。因此，PO 更以客户为导向，而 BA 通常更专注于项目。专业的业务分析师通常有资格接管产品所有者的一些任务，例如管理产品积压、建模工作流程等。

## 专案经理

* 确保产品或其部件按时交付并在预算范围内
* 管理和激励软件开发团队

在顺序模型中，项目经理负责在团队成员之间分配任务、规划工作活动和更新项目状态。在注重自我管理、透明度和共享所有权的敏捷项目中，PM 设定产品愿景、保持透明度、促进沟通、寻求开发过程中的改进，并确保团队提供更多价值每次迭代。一些人认为，在敏捷环境中不需要 PM 并担任类似角色，例如服务交付经理或 Scrum Master，但这并不完全正确。但是，如果您的公司同时运行多个敏捷项目，那么拥有专门的 PM 至关重要。他们将连接高层利益相关者需求和团队级别的日常任务执行之间的点

## 用户体验/用户界面设计师

* 将产品愿景转变为用户友好的设计
* 创建用户旅程以获得最佳用户体验和最高转化率

产品设计过程有两个方面——用户体验（UX）设计和用户界面（UI）设计。UX 部分代表思考用户与产品交互的整个旅程。因此，用户体验设计师参与诸如用户研究、角色开发、信息架构设计、线框图、原型制作等活动。反过来，UI 设计师会为产品设计直观、易于使用且令人赏心悦目的界面。UI/UX 设计师将在整个开发生命周期中陪伴您，帮助您通过功能性和引人入胜的用户体验实现业务目标，并随着时间的推移分析、评估和增强这些体验。

## 软件架构师

* 设计高级软件架构
* 选择合适的工具和平台来实现产品愿景
* 建立代码质量标准并执行代码审查

专家级软件工程师，架构师是在应用程序开发团队中制定执行软件设计决策的人。如果您处理具有复杂需求的软件产品或需要进行深刻更改的遗留软件，您将需要一个。软件架构师决定哪些服务和数据库应该一起通信，集成应该如何工作，以及如何确保产品的安全和稳定。

## 软件开发师

* 设计和稳定产品
* 解决开发生命周期中出现的任何技术问题

软件开发人员执行实际工作并编写应用程序代码。就像应用程序具有前端和后端一样，也有前端和后端开发人员。前端开发人员创建用户与之交互的应用程序部分，确保应用程序为所有人提供同样流畅的体验——无论设备、平台或操作系统如何。反过来，后端开发人员实现了应用程序的核心——它的算法和业务逻辑。经验丰富的后端开发人员不仅编写代码，还执行架构师的任务——例如，设计应用架构或设计并实施必要的集成。也有全栈开发人员。他们可以一次处理所有工作——从客户端到服务器再到数据库，以及所有需要的集成。

## 软件测试工程师

* 确保应用程序按照要求执行
* 发现功能性和非功能性缺陷

软件测试人员的工作是验证应用程序是否满足要求——包括功能性和非功能性的。功能性需求定义了应用程序应该做什么，而非功能性需求指定了它应该如何做。为了验证两者，测试工程师运行各种检查，然后分析测试结果并报告应用程序质量。他们从不同的角度验证应用程序——无论是功能、可用性、安全性还是性能（因此，有多种类型的测试）。为了跟踪执行的检查并确保测试涵盖所有需求，QA 专家可能会创建不同类型的测试文档——从测试场景到测试协议再到测试结果报告。和[经验丰富的 QA 工程师](https://itrexgroup.com/services/software-testing-qa/)设计和实施有助于在开发后期防止缺陷的质量保证流程和程序。

## 测试自动化工程师

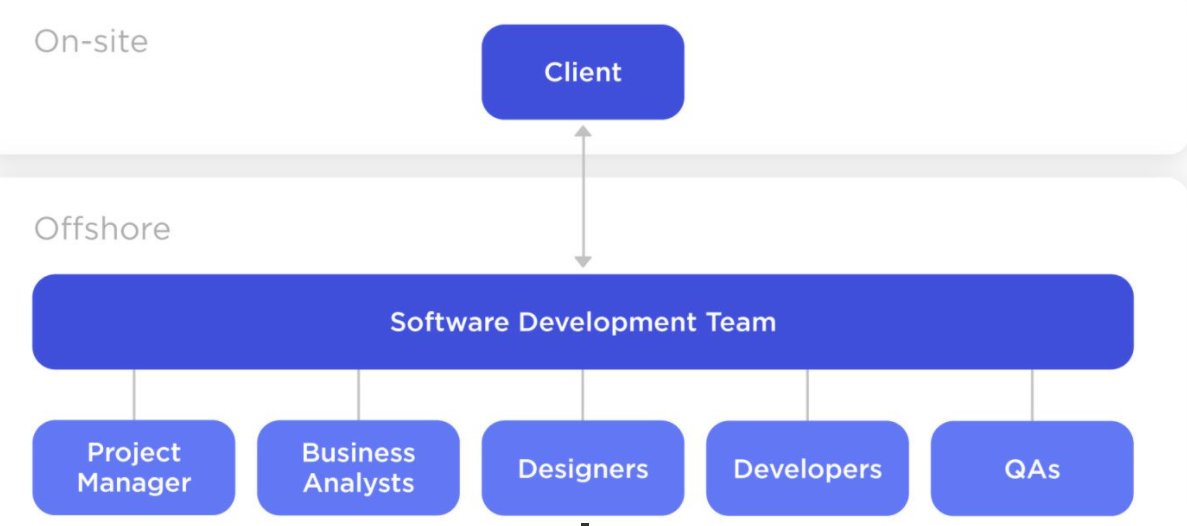
* 设计测试自动化生态系统
* 编写和维护用于自动化测试的测试脚本

[测试自动化工程师](https://itrexgroup.com/services/automated-testing/) 可以帮助您更快更好地进行测试 。为了实现这一点，他们开发了测试自动化脚本——无需任何人工参与即可提供关于应用程序质量的可靠和持续反馈的小程序。熟练的测试自动化工程师会帮助您选择应用程序的哪些部分[适合自动化](https://itrexgroup.com/blog/manual-vs-automation-testing/)，哪些部分最好手动测试。他们还将设计一个易于维护和更新的测试自动化生态系统。最后，他们将确保您的测试自动化计划以合理的成本产生尽可能多的价值。

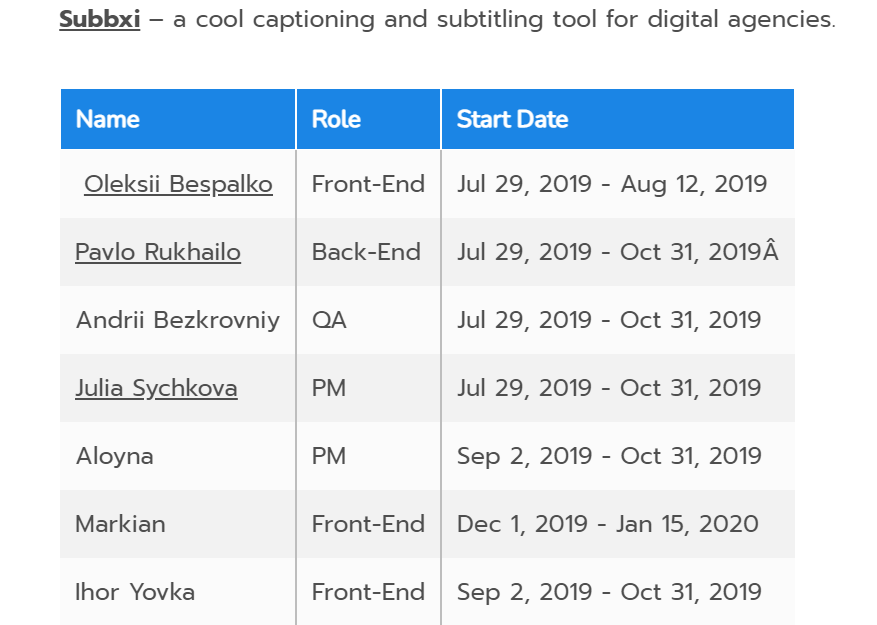
## 开发运维工程师

* 促进开发和运营团队之间的合作
* 构建 CI/CD 管道以加快交付速度

即使在敏捷环境中，开发和运营团队也可能是孤立的。[DevOps 工程师](https://itrexgroup.com/services/devops/) 充当两个团队之间的纽带，统一和自动化软件交付过程，并帮助在快速引入更改和保持应用程序稳定之间取得平衡。DevOps 工程师与软件开发人员、系统管理员和操作人员合作，在 CI/CD 的基础上监督和促进代码发布。



* **软件开发团队结构示例**



# 管理方式

# 国内

## 垂直团队组织

* 垂直团队由多面手组成。用例 分配给了个人或小组，然后由他们从头至尾地实现用例。
* **优点**  
  　　· 以单个用例为基础实现平滑的端到端开发。  
  　　· 开发人员能够掌握更广泛的技能。
* **缺点**  
  　　· 多面手通常是一些要价很高并且很难找到的顾问。  
  　　· 多面手通常不具备快速解决具体问题所需的特定技术专长。  
  　　· 主题专家可能不得不和若干开发人员小组一起工作，从而增加了他们的负担。  
  　　· 所有多面手水平各不相同。
* **成功因素**  
  　　· 每个成员都按照一套共同的标准与准则工作。  
  　　· 开发人员之间需要进行良好的沟通，以避免公共功能由不同的组来实现。  
  　　· 公共和达成共识的体系结构需要尽早在项目中确立。

## 水平团队组织

## 水平团队由专家组成。此类团队同时处理多个用例，每个成员都从事用例中有关其自身的方面。

* **优点**  
  　　· 能高质量地完成项目各个方面（需求、设计等）的工作。  
  　　· 一些外部小组，如用户或操作人员，只需要与了解他们确切要求的一小部分专家进行交互。
* 缺点  
  　　· 专家们通常无法意识到其它专业的重要性，导致项目的各方面之间缺乏联系。  
  　　· “后端”人员所需的信息可能无法由“前端”人员来收集。  
  　　· 由于专家们的优先权、看法和需求互不相同，所以项目管理更为困难。
* 成功因素  
  　　· 团队成员之间需要有良好的沟通，这样他们才能彼此了解各自的职责。  
  　　· 需要制定专家们必须遵循的工作流程和质量标准，从而提高移交给其他专家的效率。

## 混合团队组织 　　混合团队由专家和多面手共同组成。多面手继续操作一个用例的整个开发过程，支持并处理多个使用例中各部分的专家们一起工作。

## 优点 　　· 拥有前两种方案的优点。 　　· 外部小组只需要与一小部分专家进行交互。 　　· 专家们可集中精力从事他们所擅长的工作。 　　· 各个用例的实现都保持一致。

## 缺点 　　· 拥有前两种方案的缺点。 　　· 多面手仍然很难找到。 　　· 专家们仍然不能认识到其他专家的工作并且无法很好地协作，尽管这应该由多面手来调节。 　　· 项目管理仍然很困难。

## 成功因素 　　· 项目团队成员需要良好的沟通。 　　· 需要确定公共体系结构。 　　· 必须适当地定义公共流程、标准和准则。

## 阿里软件开发管理方式案例

* **服务体系**



* **算法团队能力**

需要具备 4 项能力：连接-生产-传播-服务。

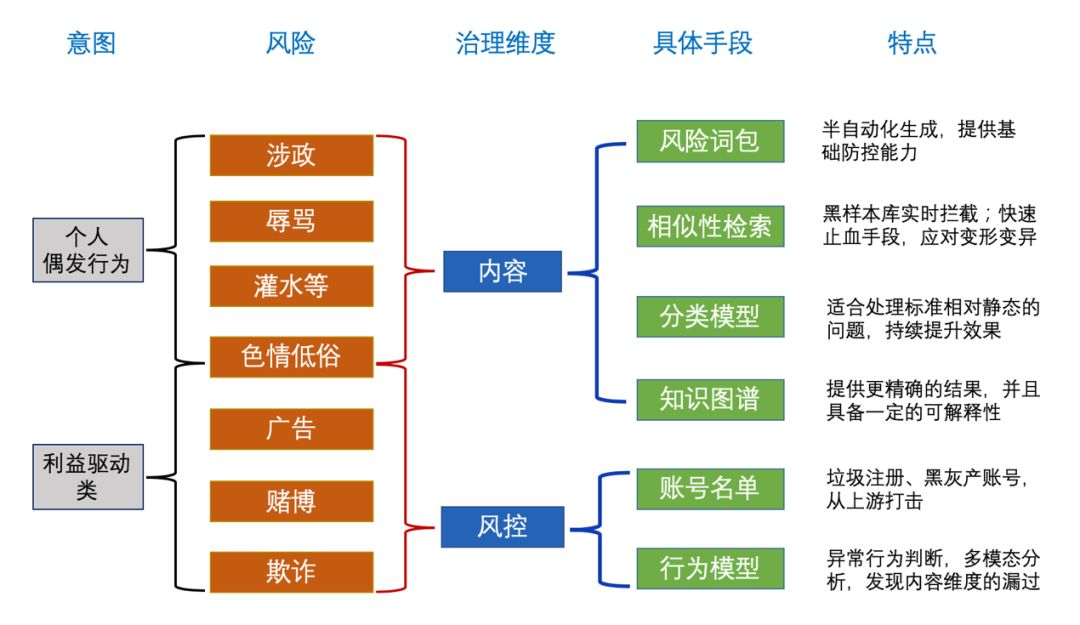


* **工作模式——正确地做事**

**四个在线化**



* **全局视角的UGC风险管控思路**



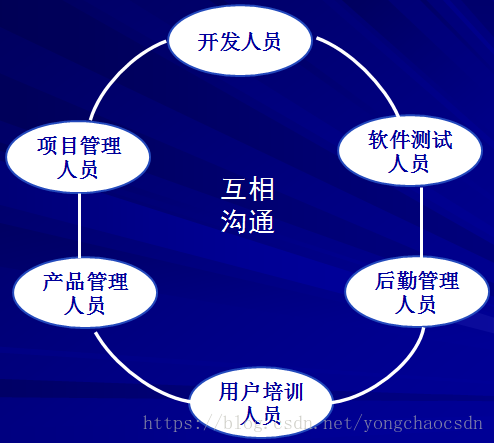
* 绩效的考核



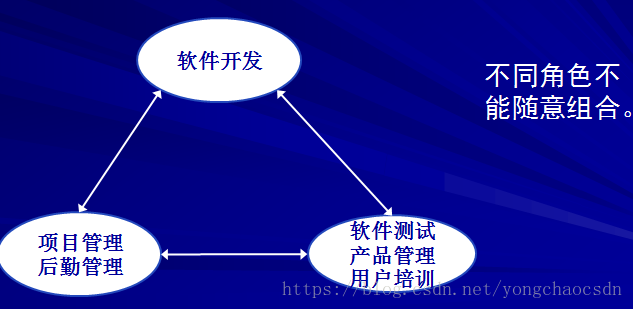
# 国外

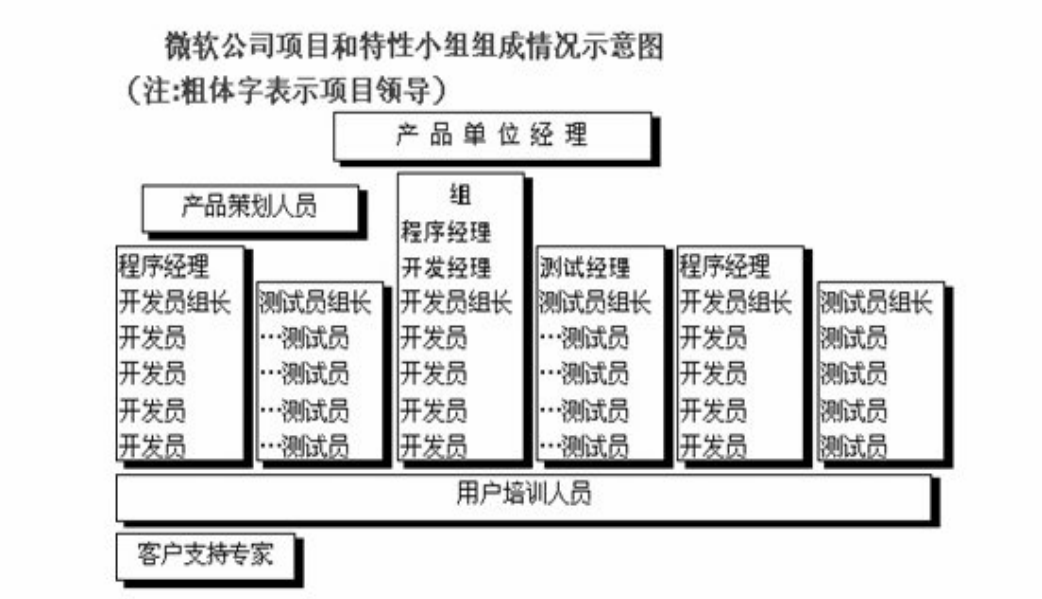
## 微软

微软的一个软件产品项目团队通常由一个“产品单元经理”（Product Unit Manager）领导，包含有不同的角色。



在小型项目中，可以简化团队的组成，也就是说，可以把一些角色组合在一起。





* **同步--稳定法开发模式**

微软公司的项目运作方式是把项目分成若干个子项目,并根据功能领域分组同时进行的平行推进工作方式,这种方法集中了里程碑和每日构造这些关键的概念。微软典型的项目管理(项目的生命周期）包括三个阶段:

计划阶段完成功能的说明和进度表的最后制定;。开发阶段写出完整的源代码;

。稳定化阶段完成产品,使之能够批量生产。

* **MSF的八个基础原理**
* 推动开放式沟通
* 为共同的前景而工作赋予小组成员权力
* 建立清晰的责任和共同的职责
* 关注交付业务价值
* 保持灵巧，预测变化
* 质量投资
* 学习所有的经验