

软件开发中人员管理

目录

- 信息系统开发人员分类;
- 各类人员的职责;
- 各类人员之间的关系;
- 各类人员在开发中的作用;

信息系统开发人员分类及其职责概述

信息系统开发是一个系统的复杂的工程，涉及到许多各类人员的共同参与，是集体的成果。搞清楚软件或信息系统开发中需要那些人员？各类人员都有什么职责？是很有必要的。

信息系统开发人员分类（1-7）

- 1、项目投资人： 高层领导，批准并支持系统的管理人员；
- 2、项目经理： 负责系统总体职责的管理人员；
- 3、项目领导： 技术队伍的协调者；
- 4、技术人员： 所有指派到项目中的开发人员和职员；
- 5、最终用户： 那些作为他们的日常工作来操作该系统的人；
- 6、最终用户开发者： 来自最终用户组参加开发队的非系统人员；
- 7、系统设计师： 总体技术设计人员；

信息系统开发人员分类（8-14）

- 8、应用分析员： 复审最终用户需求并定义系统说明的人；
- 9、数据库管理员： 数据库的管理人员；
- 10、数据管理员： 负责命名和控制数据元素名和内容的人；
- 11、界面设计者： 决定最佳人—机接口方法和风格的技术人员；
- 12、网络设计师： 负责通过数据通讯工具和协议把所有系统成份集成起来的专业技术人员；
- 13、文档人员： 开发用户手册、帮助程序和系统的技术说明的专业技术人员；
- 14、系统测试人员： 进行系统单元测试和联合测试的人员。

1、项目投资人

- 1、在所有成功的管理信息系统项目中，总是有一个在机构中的高层领导出于某种原因想要建立这个新系统。
- 2、项目投资人负责标识系统的战略价值并把可以为整个企业带来利益的事务或服务与该技术联系在一起。
- 3、项目投资人一般站在高层来操作，因而看到的是由机构的高级管理组织所定义的“大图”。这也同样适合于部门系统，只是项目投资人需要把其视野限制到以部门作为开端。
- 4、项目投资人设定系统的托管和范围。这一角色通常由一名高级管理人员担任，他完全了解企业战略并且也对其他高级部门经理进行访问。
- 5、项目投资人还能够把该项目的成功转变为使用了客户/服务器技术的新型的系统应用。
- 6、项目投资人大多数情况下只在开始阶段和批准阶段参与。但项目投资人也要出席一些具有里程碑意义的会议并且要看看主要的交付演示，这有利于把项目队职员集中于他们对结果的承诺上。

2、项目经理

- 1、项目经理负责对项目的资源和人员进行管理。担当此角色的人负责按时间计划和在项目投资人预算并批准的财政资源条件下交付系统。
- 2、项目经理决定优先次序并解决项目范围内任何个人间的优先序和设计矛盾。
- 3、项目经理把应用的战略价值翻译成系统必须实现的核心目标的定义。
- 4、由于项目投资人一般均系高级管理人员，项目经理则直接与项目组成成员打交道。
- 5、项目经理主要负责各种项目成份的集成，并保持项目在设计与开发阶段始终聚焦于它对企业的价值上。
- 6、项目经理在开始阶段需要极大的投入，在后续阶段仍要保持对项目状态和出现的困难的了解。

关于项目经理，后续课程还有详细讨论。如素质、能力、职责等。

3、项目领导

1、项目领导为每个人分配具体的任务，并把由项目经理确定的总体的项目交付任务分划成一系列具有其止日期的离散活动。

2、详细的项目计划的制定通常也是项目领导的职责。项目领导应该时刻准备了解每个成员的工作以及在任何给定的时候项目队所面临的技术挑战的实质。

3、如果一个开发人员或另一个开发成员比交付计划完成得晚或遇到了技术上的难题，他或她首先要向项目领导请求帮助解决问题。

4、项目领导对于保证每个队员的工作的协调也负有责任。

5、他或她也要负责保证由全体项目职员完成工作的质量水准。

6、项目领导按计划把开发现状向项目经理汇报，通常还负责与将要使用该系统的最终用户进行通信。

7、任何对最终用户有影响的二次工程需求都应该首先告知项目领导，就这些问题与用户的沟通项目领导负有主要责任。

4、技术人员

1、在任何项目中，必须有人去做实际工作。在信息系统开发中，这一职责落在了技术人员的肩上，他们通常是软件开发人员。

2、对于信息系统开发中的每一部分而使用的每个具体的技术产品，都必须从项目开发队伍中指派一个**主要的技术人员**。这样可以保证有一个专人负责了解这个产品并可在需要的时候把他的知识传授给其他队员。

3、技术人员变成了信息系统开发中某个方面的内部专家。这些方面包括：SQL语法、系统集成、应用软件、设计与开发工具或SQLSERVER 数据库管理系统。

4、信息系统开发提供的特色的复杂性使得在一些主要环节 上检查对那一技术的适当使用非常重要。若不强调这一点，一些只要发现即可解决的小问题就可能滋变成对整个项目的严重威胁。

5、由于很多成功的信息系统项目使用了较小的开发队，一个开发人员就有可能成为承担多项技术的技术人员，作为一名开发人员，你应该能够标识你有责任支持哪些技术，并了解在信息系统开发技术的其他方面你可向谁求助。

5、最终用户

- 1、任何参加你的信息系统项目的人都应该有一个明确的认识：谁是最终用户。最终用户负责实际接手这个应用，并用此为企业在将来创造价值。
- 2、这点并不是总象听起来那样直观。当不只作为部门的试验时，信息系统意在成为人的工具，只有使用得当，有效的工具才能发挥得最好。
- 3、把这条原理应用到你的系统中去的具体办法是，了解你的最终用户、他们的需求及目标。若开发一个客户/服务器系统，把它交给最终用户，然后抱怨用户使用不当，这是毫无意义的。
- 4、最终用户有他们自己的组织需求，由这些需求驱动的系统交付正是必须由你的客户/服务器系统实现的。
- 5、有这样一个不幸的倾向：系统分析员相信他们对用户需求的理解比用户本人更好，由于对用户的忽视，他们失去了最终用户的看法。
- 6、项目成功取决于你的系统如何正确地识别并满足了最终用户的需求，而不是用户如何好地掌握并操作你的系统。
- 7、他们是你的客户，你的开发工作的成功取决于你如何好地使你的系统投入工作。你要理解用户，而不是让用户理解你。不管系统的设计与实现有多好，只有他们——与系统共同工作的最终用户才能实现收益。

6、最终用户开发者

- 1、大多数最终用户要么是过于忙，要么是对把很多时间花在系统开发过程中不感兴趣。要想真正成功，你的客户/服务器项目需要足够多的用户输入和反馈。
- 2、当最终用户额外担当了开发者的角色时，他们就主动变成了连接已完成产品的最终用户和负责建立应用的技术人员这间的桥梁。
- 3、最终用户开发者的第一个职责是揭示将来系统的最有用的特色的精确内涵，第二个职责是把尚在开发中的系统向其他最终用户作宣传。
- 4、最终用户开发者担当起两组人之间的翻译。最有效的客户/服务器系统是利用某种形式的反复开发建立起来的，这种反复式开发是指在整个开发过程中借助用户反馈快速地进行改进和增强。最终用户开发者可提供大量的这类反馈。

- 5、然而，这个过程还应包括另外两项功能。正像你将会在你的客户/服务器项目中扯到的那样，系统有可能会改变最终用户的业务方式。新的机能就是新的机遇，它将迫使用户现状做一次有益的变革。最终用户与软件开发人员之间的通讯也不只限于一种方式。它是双向式的，而且最终用户开发者可以起到一个非常有价值的作用，那就是使最终用户了解新系统可能提供的新机遇。
- 6、最终用户对新的客户/服务器应用熟悉得越快，也就越有可能采用它。最终用户开发者从本质上担当起了预先培训其他最终用户的职责，使得尚在开发之中的系统看起来更加熟悉和不可怕。这个职责是难于由系统人员承担的，因为最终用户一般不认为系统人员能够真正理解他们，或者是他们中一员，为使这一角色更为有效，应该选择那些对最终用户组整体较有影响的最终用户作为最终用户开发者。

7、系统设计师

1、系统设计师的角色不仅仅是设计一个客户/服务器应用。系统设计师必须把企业和计算环境作为一个整体来考虑。在这样的上下文环境中，系统设计师定义客户/服务器系统中各元素如何协作。系统体系结构包括硬件、软件和系统的集成。对于你的客户/服务器项目，系统设计师要负责环境的总体技术设计、为各种组件选择合适的工具、应用与所有系统资源的集成。

2、在项目开始阶段，系统设计师与项目经理和项目投资人在一起工作以保证项目在技术和操作上的长远目标。系统设计师应该是非常熟悉技术本身且保证项目选定的技术具有长远价值。技术上的长远价值在于可以充分利用现有资源，或者在技术和经验上的投资必须在将来和现在都是有用的。一个长远性和系统体系结构应该是这样一种体系结构：它采用的技术在将来不会变得过时或者难懂。

3、系统设计师也要负责保证项目在技术上是可行的。客户/服务器计算机工业中快速变化的领域之一。系统设计师负责保证被选择的每项技术都确实能够提供所需的服务，它们将被集成为一个有效的总体方案。不能投入运行的系统是无法实现重要的价值的。

8、应用分析员

1、站在与系统设计师相对的另一端，应用分析员负责开发一个特定的客户/服务器应用将要完成的功能的说明。应用分析员通常负责为现有工作过程和系统做文档，并且与用户共同工作来确定新系统实际将要做什么。应用分析员定义应用时总是站在用户或业务的角度，保证系统完成正确的功能，满足用户的商务目标，而不是开发一个详细的程序设计的说明。应用分析员与用户一道弄清楚新系统对于当前的事务实践将会产生什么样的影响。

2、应用分析员审查用户环境并让用户用自己的语言解释他们的作业。应用分析员再将这个描述翻译给其他人去阅读包括开发队成员，开发队正是从这个文档理解用户的需求。这种理解再翻译成可用于创建软件应用本身的程序设计说明。

3、应用分析员从第一代计算机开始就是软件开发过程的有机部分。用于把应用说明文档化的技术已有很多，应用分析员有责任使用这些技术为你的具体的客户/服务器系统做文档。

9、数据库管理员

1、DBA的责任是开发逻辑和物理数据模型。这些数据模型详细叙述每个必须被用来支持应用的数据库对象，包括所有的表、视图、存储过程、规则、缺省、索引和触发器。DBA通常对于如何创建数据库具有决断权，对于保证所有数据库对象的创建和数据库对正在开发中的应用的支持也负有主要责任。由于数据库应用的服务器方面与客户方面之间固有的分离本质，因而即使采用适合数据库设计的所有准则也完全有可能使开发出的数据库结构无法支持应用。数据库管理员与开发队共同努力，通过迭代式的逻辑和物理数据模型，保证数据库设计将支持应用。

2、在物理数据模型实现这后，DBA与开发队要对系统性能进行复审并根据查询的实际性能需求修改数据库以加速数据检索的速度。

3、在大多数机构中，数据库管理员处在负责管理很多数据库应用的中心位置上。在这种情况下，数据库管理员在责任保证数据库对象，如存储过程，是经过测试和行成文档，并且是与整个机构范围的标准相符。数据库管理员应该知道那些已有的、可能会被开发队使用或修改的存储过程或其它数据库对象。

4、DBA通常在你的SQL Server上有一个SA帐号，并且负责建立所有用户、帐号和权限，还要为能充分利用SQL Server在基上运行的硬件平台而进行物理数据模型的设计。SQL Server体系结构中有很多方面不为软件开发人员所见，但却影响着应用的性能。DBA对此以及库安全和数据完整性均负有主要责任。¹⁵

10、数据管理员

1、数据管理员是的职责更加注重于数据类型定义的整体标准以及对表及列的命名。数据管理员确保标准编码规则规范的使用，经常把在机构别处所做的工作借鉴过来，或者使用适合的标准化组织，如IEEE、OSF、CCITT和GOOD HOUSEKEEPING 等认可的代码表。

2、数据管理的另一职责是保证多个数据库是可映射在一起的，并且相近的是术语在整个数据库中应具有完全相同的含义。数据字典或数据目录使用的定义规范也是数据管理员的职责，尽管数据字典的创建是数据库管理员的任务。

3、许多项目不明确指定数据管理的职责，DBA完成所有事；对于大型项目或大机构的小项目，数据管理必须由专人负责。

4、对于信息系统开发，数据管理问题是集成能否成功的中心问题。集成经常失败的原因不是技术上不可能而是数据匹配不起来。不具有外部引用或外键的特定应用代码表经常导致这种情况的根源。

11、界面设计者

- 1、随着应用程序向图形用户界面的过渡，对界面设计者这一新角色的需求便产生了。
- 2、对有效的人机界面设计与开发的强调曾使传统应用受益。用户界面设计者需要决定哪些屏幕需表现给用户，并保证可视化布局和设计规则在这些屏幕上的体现。
- 3、随着多媒体应用更多地变成信息系统的一部分，适当地应用声音、色彩、图形和表格的职责都落在用户界面设计的肩上。
- 4、除了定义应用的外观外，用户界面设计者需要决定哪些界面选项对于用户想要由应用完成的给定的任务最为适合。
- 5、菜单布局 and 结构、任务选项和特色、应用的规定如键盘的简捷操作和快键等都是用户界面设计者的职责，他们要保证你的技术上的有效的项目在外观表现上亦佳，并且也与最终用户的工作方式相适应。

12、网络设计师

1、信息系统中把环境中的通讯和网络服务当成是集成方案中的一个关键。

2、为了确保通信和网络设计能够支持数据的吞吐量需求，必须指定一个网络设计师负责端-端联结的网络拓扑的规划和设计。

3、设计师负责标识把客户和服务器连接起来所需的所有软件和硬件，从而使应用可透明地在通讯服务和协议的复杂环境下来工作。

13、文档人员

1、尽管文档的开发长期以来一直被当作正规开发的系统的重要部分，而软件开发的现实则是经常把文档开发置于脑后。

2、建立清晰和内容丰富的联机帮助是文档人员的职责；培训和通讯技术的进展如多媒体技术也对文档的定义产生了影响。

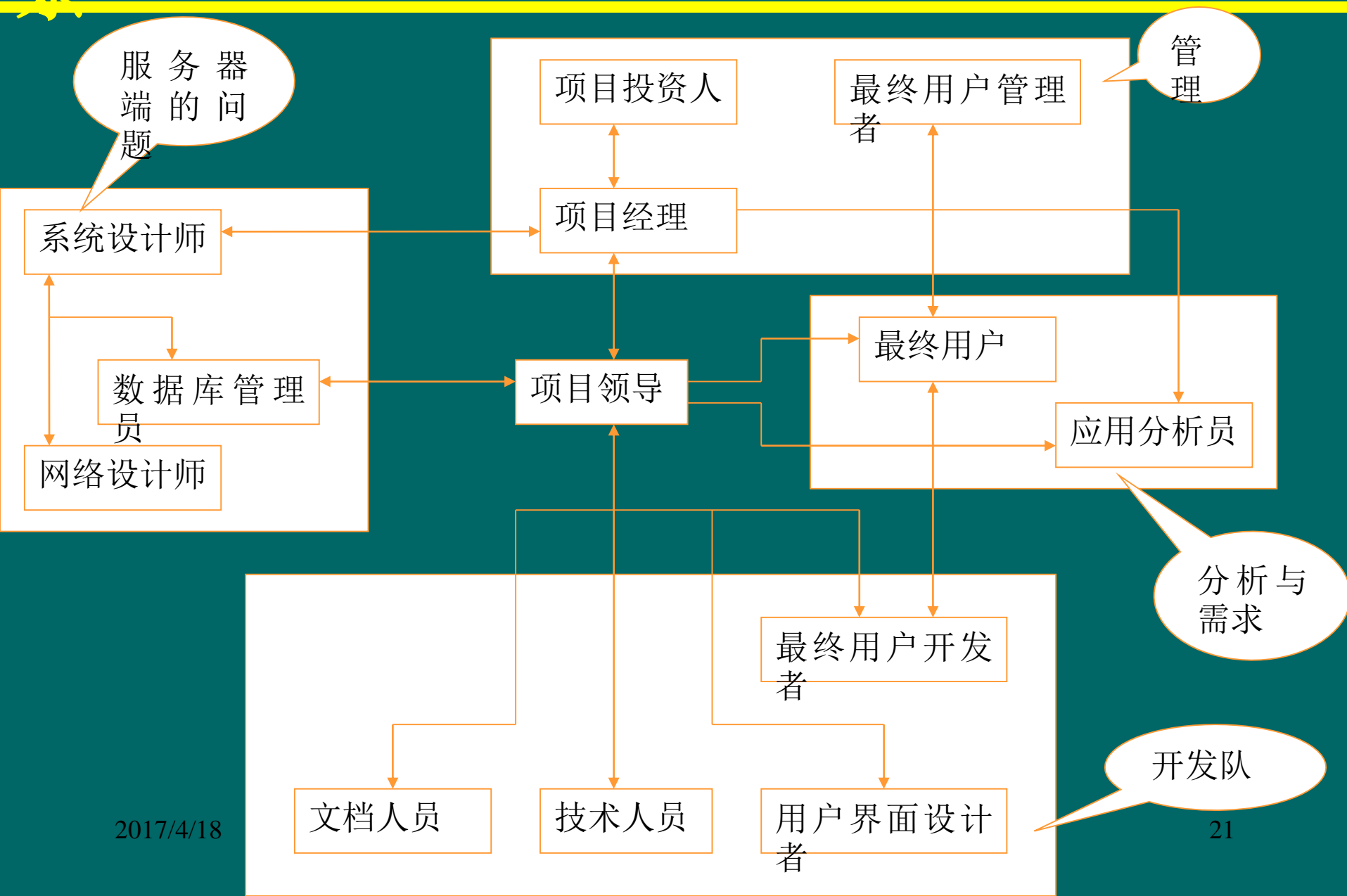
3、做文档不仅是为了你建立的系统的最终用户和操作员。对于提高可维护性并使新的特色能被快速地增加进来，把软件开发过程文档化也将是非常有用的。

4、文档人员不是简单地把系统写下来。文档人员还需要懂得人们如何学习，了解应用的结构，只有这样才可使用户和将来的开发人员的工作更为有效。

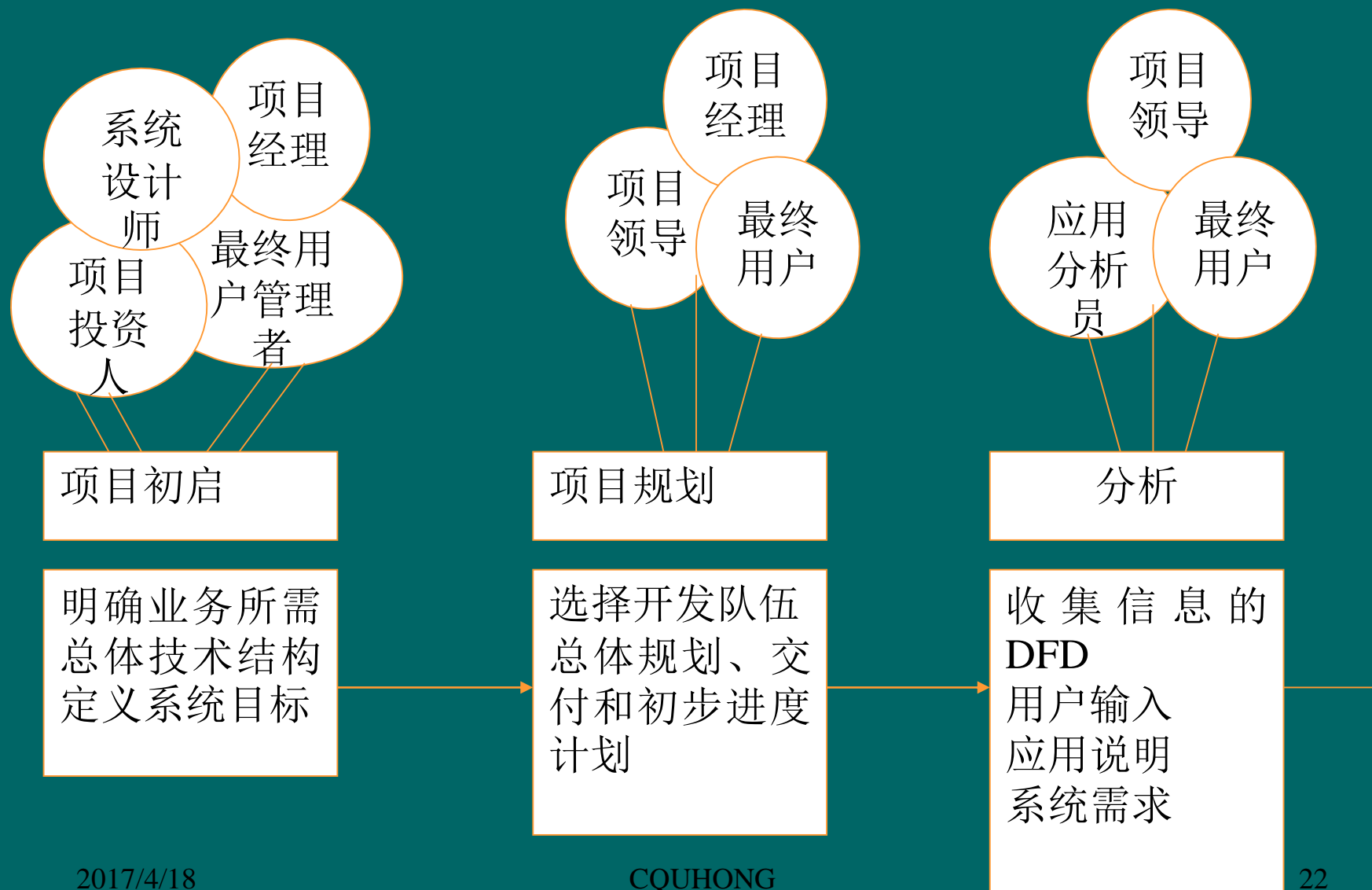
14、测试人员

- 1、系统必须是通过严格测试之后较给用户的。
- 2、开始是程序员测试编开发边测试；
- 3、专人单元测试；
- 4、专人联合测试；
- 5、用户测试。

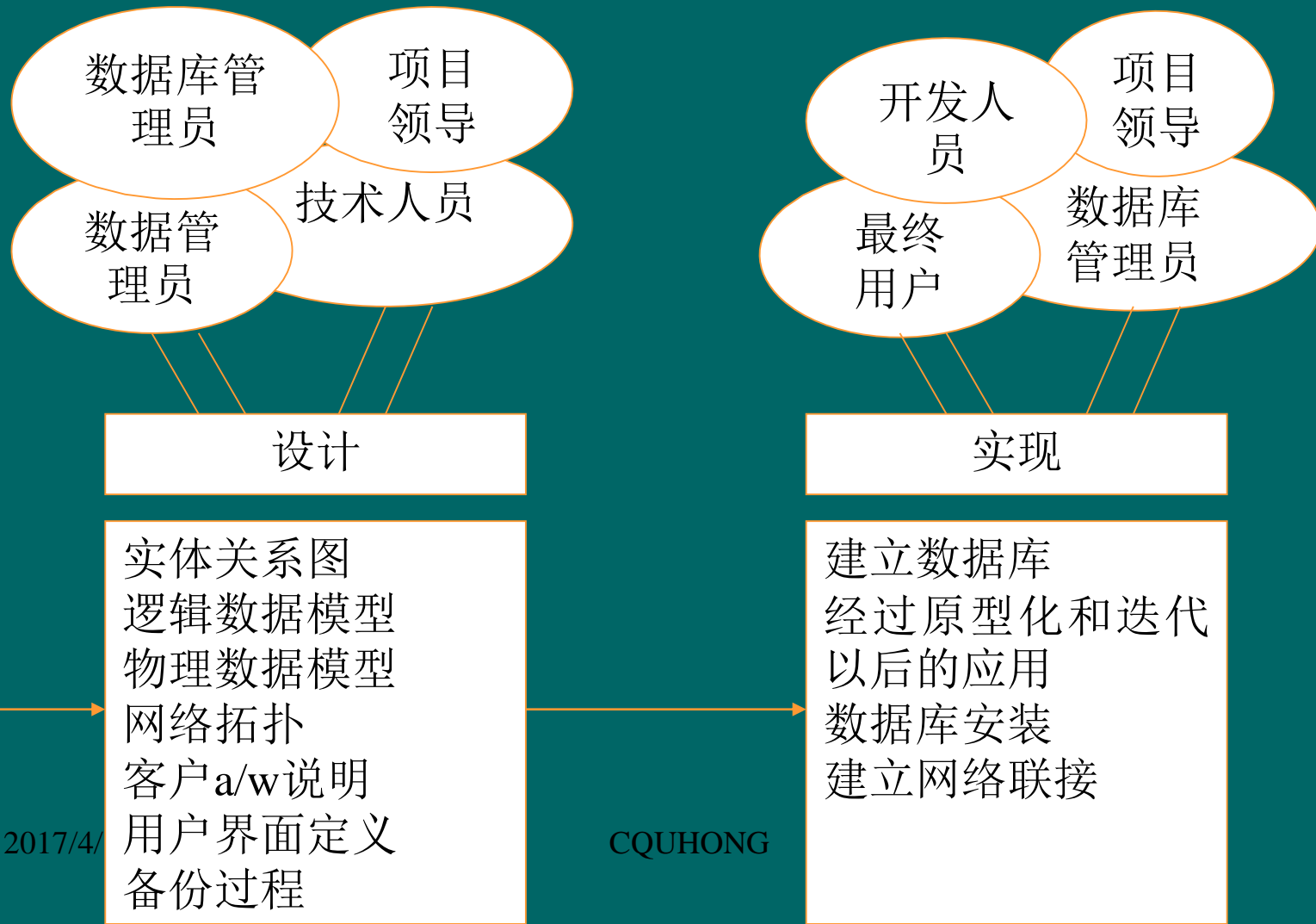
各类人员之间的关系



各类人员对开发过程的影响



各类人员对开发过程的影响（续）



软件项目组

软件工程小组的组织方式：

- 民主分权式（Democratic Decentralized, DD）：这种软件工程小组没有固定的负责人。任务协调者是短期指定的，之后就由其他协调不同任务的人取代。问题和解决方法的确是由小组讨论决策的。
- 控制分权式（Controlled Decentralized, CD）：这种软件工程小组有一个固定的负责人，他协调特定的任务及负责子任务的二级负责人关系。问题解决仍是一个群体活动，但解决方案的实现是有小组负责人在子组之间进行划分的。子组个人间的通信是平行的，但也会发生沿着控制层产生的上下级的通信。
- 控制集权式（Controlled Centralized, CC）：顶层的问题解决和内部小组协调是由小组负责人管理的。负责人和小组成员之间的通信是上下级式的。

软件项目组

计划软件工程小组结构时应该考虑的因素：

- 待解决问题的困难程度；
- 要生成的程序的规模，以代码行或功能点来衡量；
- 小组成员需要待在一起的时间（小组生命期）；
- 问题能够被模块化的程度；
- 待开发系统所要求的质量和可靠性；
- 交付日期的严格程度；
- 项目所需要的社交性（通信）的程度。

软件项目组

- 因为集中式的结构能够更快地完成任务，因此最适合处理简单问题。而分散式的小组比起个人而言能够产生更多更好的解决方案，因此这种小组在处理复杂问题时成功的可能性更大。因为CD小组是集中式地解决问题，所以CD或CC小组结构能够成功地用来解决简单的问题。而DD结构则适合于解决难度较大的问题。
- 因为小组的性能与必须进行的通信量成反比，所以如果子组很容易协调的话，很大的项目最好采用CC或CD结构的小组组织方式。

软件项目组

- DD小组结构最适合于解决模块化程度较低的问题，因为它需要更多的通信。如果有可能要较高的模块化程度，则CC或CD结构更加合适。
- CC和CD小组已被发现能够产生比DD小组更少的缺陷，但这与小组所采用的质量保证活动密切相关。分散式结构通常需要比集中式结构更多的时间来完成一个项目，但是如果要求高社交性，它是最合适的。

软件项目组

软件工程小组的组织范型：

- 封闭式范型：按照传统的权利层次来组织小组（类似CC小组）。这种小组在开发与过去已经做过的产品类似的软件时十分有效，但这种封闭式范型下难以进行创新式的工作。
- 随机式范型：松散地组织小组，并依赖于小组成员个人的主动性。当需要创新或技术上的突破时，按照这种随机式范型组织的小组很有优势。但当需要“有次序的执行”才能完成工作时，这种小组组织范型就会陷入困境。

软件项目组

- 开放式范型：试图以一种既具有封闭式范型的控制性、又包含随机式范型的创新性的方式来组织小组。工作的执行结合了大量的通信和基于小组一致意见的决策。开放式范型小组结构特别适合于解决复杂问题，但可能不象其他类型小组那么效率高。
- 同步式范型：依赖于问题的自然划分，组织小组成员各自解决问题的片段，他们之间没有什么主动的通信需求。