

软件工程实践
Rational 用户 组



uml.org.cn
UML 软件工程组织

软件文档体系讲座



UML 软件工程组织

郑明华



Evolve by case

议题

- 文档体系与软件工程化
- 文档体系标准化
- 文档体系的组织结构
- 文档体系实际应用中的技巧
- 文档设计的实用技巧



什么是文档？

- 文档是指某种数据媒体及其中所记录的数据，它具有词义性，并可由人或机器阅读。
- 软件文档则是指在软件开发过程中，用来表示对活动、需求、过程或结果进行描述、定义、规定、报告的任何书面的文字或图示信息的文档。



什么是软件文档体系

- 文档体系就是将有关于某一应用的所有文档进行合理管理的流程、规范与制度。
- 文档体系管理的重点就是各类文档。
- 软件文档体系是指与软件开发、应用、维护等相关的所有文档管理的流程、规范与制度。它管理的重点在于和软件相关的文档。



文档体系应用时的常见误区

- 文档体系就是文档类型描述
- 文档体系就是文档内容格式



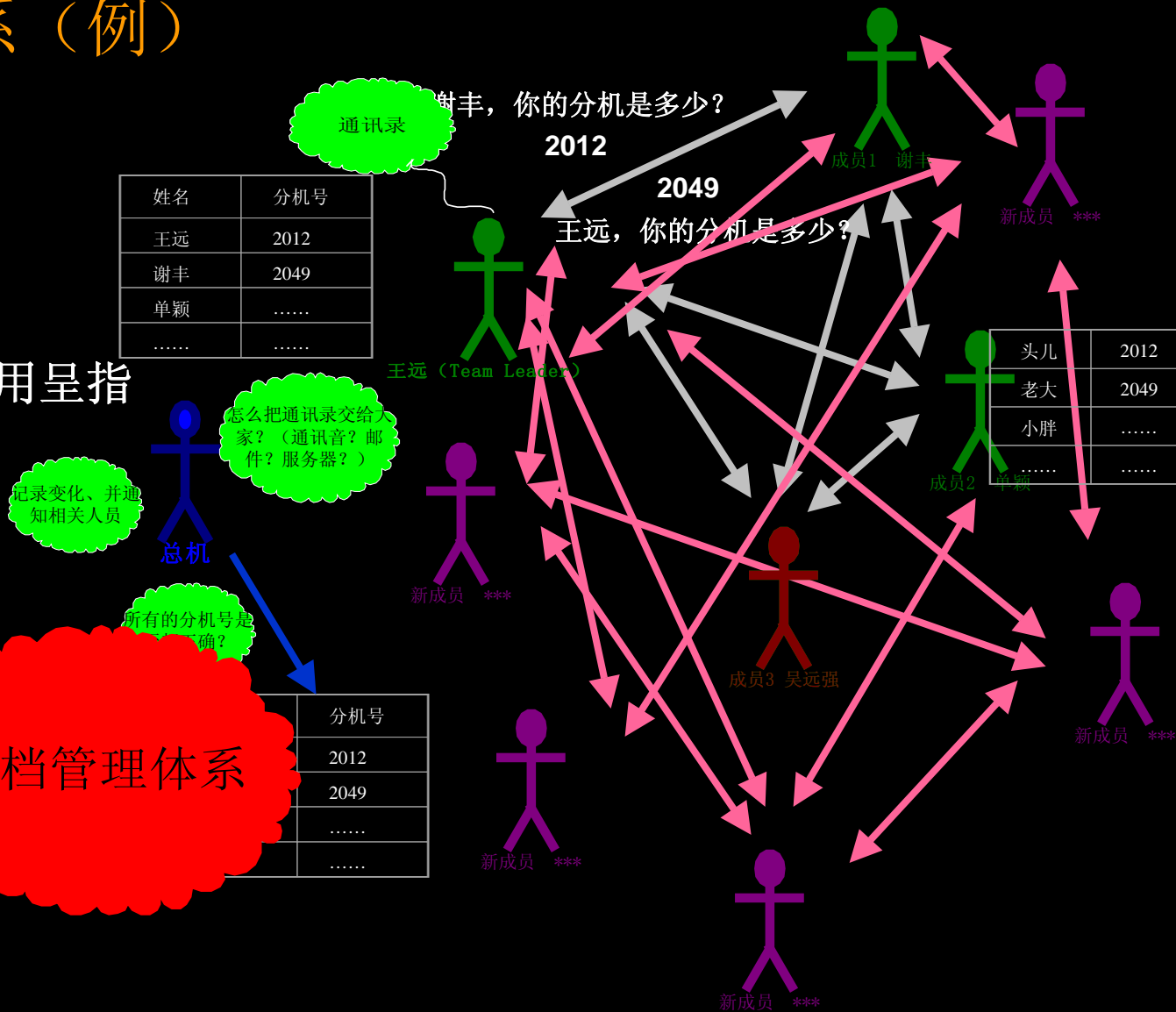
文档和文档体系（例）

■ 通讯问题

- ▶ 电话分机号
- ▶ 通讯费用 C_n^2
 - ▶ N 扩大，通讯数级扩大

■ 文档

- ▶ 文档的产生
- ▶ 文档的格式
- ▶ 文档的书写
- ▶ 文档的评审
- ▶ 文档的发布
- ▶ 文档的变更管理



王远是2012，李涛是2038，小强是2055.....

软件文档的作用

1. 是相关人员沟通工作的桥梁；
2. 提高软件开发过程的能见度；
3. 提高开发效率；
4. 可以做为阶段性工作成果提交，做为本阶段结束的标志；
5. 执行下阶段工作的参考与依据；



软件文档的作用 (Con.)

6. 检验与验收工作成果的标准；
7. 有利于后期的调试、维护工作的进行；
8. 有利于用户了解软件的功能、性能等各项指标。成为用户购买或验收的重要依据；
9. 是软件开发组织的财富；



软件文档体系与软件工程

- 软件文档体系是软件工程的一个重要组成部分。
- 软件文档体系体现了软件工程的过程与作用。
- 软件文档是将软件产品抽象的开发、管理过程具体化的重要手段。
- 软件文档在软件开发的整个生命周期中不断的产生和完善。



软件工程标准与规范的产生背景

- 解决软件危机，仅靠技术措施是办不到的，它更需要有先进的管理措施。60年代后期软件工程学的出现。
- 人们更注重项目管理和采纳形式化的标准和规范，并以各种生命周期模型来指导项目的开发进程。
- 标准与规范作为软件工程四要素之一
- 世界上第一个软件工程标准是由美国军方制定的
- GJB437—88 《军用软件开发规范》是我国第一个软件工程国军标



我国软件工程标准现状

- 软件工程标准化工作80年代初开始起步。
- 现已颁布国标22项、国军标约15项，另外航空、航天、电子、机械等部门也基于行业软件管理的需要分别制定了若干行业标准，国内大型项目也制定了自己的软件工程标准规范。



软件工程标准化的意义

- 标准提供的是最适合的或最佳方案，从而避免重复前人的失败；
- 使软件开发机构中的工程技术人员协调开展工作；
- 可提高软件的生产率；
- 可提高软件的可靠性、可维护性与可移植性；
- 有利于缩短软件开发周期、降低软件产品的成本和运行维护成本。



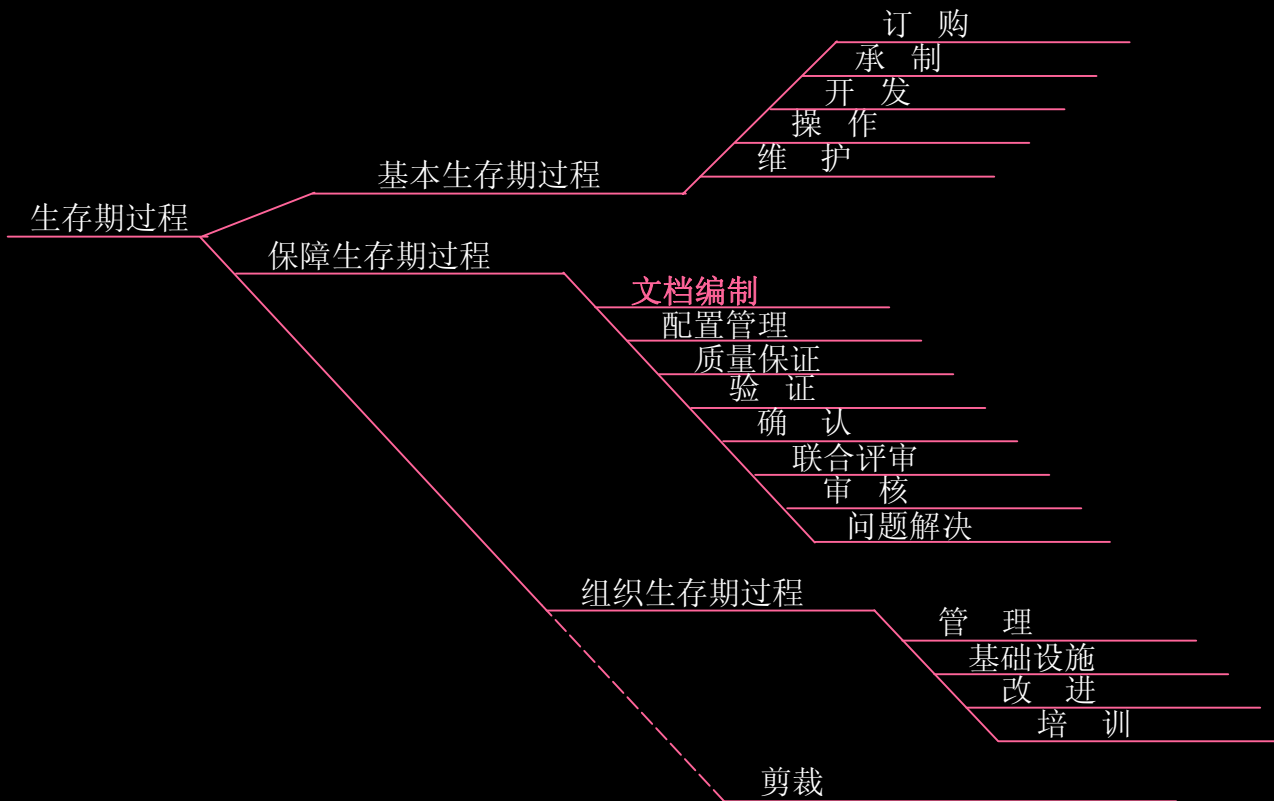
软件工程标准的级别

- 国际标准
- 国家标准
- 行业标准
- 企业标准
- 项目标准



文档在软件生存周期中的地位一

ISO/IEC 12207 之软件生存周期结构



生存期过程的结构



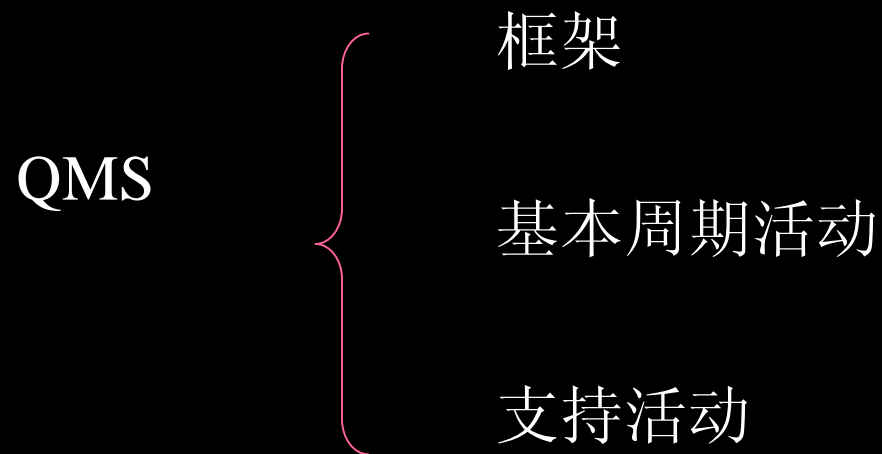
软件文档在QMS中的地位

- QMS: Quality Management System—质量管理体系
 - ▶ ISO9000-3: 质量体系—应用ISO9001进行开发、供应与维护软件的指南。
 - ▶ 是各生产机构中的一种机制，它通过协调和控制功能以求尽可能经济的达到产品/服务所要求的质量。
 - ▶ QMS是ISO9000-3在各个具体项目开发过程中的实际应用。



软件文档与QMS （续）

■ QMS的基本结构



软件文档与QMS （续）

■ 框架

- ▶ 管理职责
- ▶ 质量体系
- ▶ 内部质量体系审核
- ▶ 纠正措施



软件文档与QMS （续）

- 生命周期活动
 - ▶ 需求规格说明
 - ▶ 开发策划
 - ▶ 质量策划
 - ▶ 设计与实现
 - ▶ 测试和确认
 - ▶ 验收
 - ▶ 复制
 - ▶ 交付
 - ▶ 安装
 - ▶ 维护



软件文档与QMS （续）

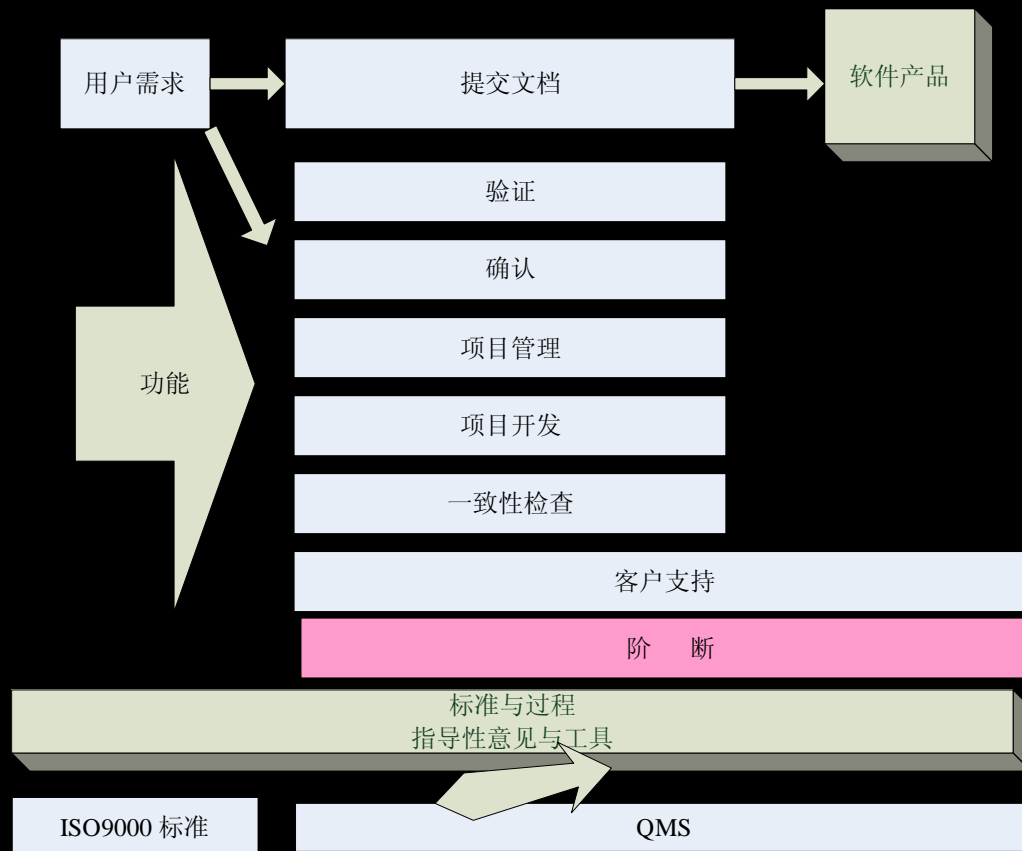
■ 支持活动

- ▶ **文档管理**（文档类型、文档的批准和发布、文档更改）
- ▶ **配置管理**（配置管理计划、配置管理活动）
- ▶ **质量记录**（度量、规则、惯例和约定、工具和技术、采购、配套的软件产品、培训）



软件文档与QMS（续）

■ QMS示例图



软件文档与QMS（续）

- 软件文档存在于QMS的方方面面
- 软件文档并不局限于软件生命周期
- 软件文档！ = 开发技术文档
- 软件文档 = 管理性文档 + 技术性文档



文档编制过程中的常见误区

- 文档中都是一些套话、缺乏有用的内容。
- 写文档太花费时间、得不偿失。
- 文档格式中很多项都不知道是什么意思，不会写文档。
- 想写的内容在文档格式中找不到相应描写的位置。
- 与其看文档，不如直接找相关人员咨询。
- 文档谁写的都一样。
- 编写文档就是浪费时间，作用是应付领导、用户。
- 习惯于补文档，而不是写文档。
- 技术文档比管理文档重要的多。



解决文档误区的策略

- 提高对文档的认识和理解
- 创造有效文档体系结构
- 制定合理的文档结构
- 提高文档编制人员能力
- 有效的辅助工具
- 良好的企业文化
- 文档格式标准化、文档管理规范



软件文档标准化

- 为什么要标准化？
 - ▶ 经验的总结
 - ▶ 规范化
 - ▶ 交流的基础
 - ▶ 有效性



文档编制过程相关标准

■ GB

- ▶ 软件文档管理指南
- ▶ 计算机软件产品开发文件编制指南 GB 8567—88
- ▶ 计算机软件需求说明编制指南 GB 9385—88
- ▶ 计算机软件测试文件编制规范 GB 9386—88

■ GJB

- ▶ 武器系统软件开发文档 GJB 438A—97
- ▶ 军用计算机软件文档编制格式和内容 GJB 1566—92



文档编制过程相关标准（续）

- 美军标MIL-STD-498 《Software Development and Documentation》
 - ▶ 发布于1994年
 - ▶ 可用于嵌入式系统和信息系统开发
 - ▶ 可用于面向对象开发
 - ▶ 被采用为新的国标的基础
- RUP的文档体系
- CMM各级中的文档体系



MIL498文档

- 软件开发计划
- 软件测试计划
- 软件安装计划
- 软件移交计划
- 运行概念说明
- 系统/子系统需求规格说明
- 接口需求规格说明
- 软件需求规格说明
- 软件设计说明
- 数据库设计说明



MIL498文档（续）

- 软件测试说明
- 软件测试报告
- 软件产品规格说明
- 软件版本说明
- 软件用户手册
- 软件输入/输出手册
- 软件中心操作员手册
- 计算机操作手册
- 计算机编程手册
- 固件保障手册



如何贯彻执行这些标准

- 面对如此众多的标准规范，该怎么办？
- 各标准之间是什么关系？
- 所有的我都要精通、了解吗？
- 从哪儿开始贯彻执行
- 一步到位还是先抓重点



使用标准规范的重点

- 使用标准规范时需要注意的是
 - ▶ 不要盲目完成所有要求的文档内容
 - ▶ 选择最有效益的文档并形成本公司规范
 - ▶ 不要为完成文档而完成文档
 - ▶ 小结：标准都是用来参考的，而不是用来迷信的。企业在参考业界推荐的标准或规范时，要舍弃那些听起来很先进但是对本企业无益处的东西，选取对企业有实用价值的东西。



标准规范的使用

- 规范的指导性与必要性
 - ▶ 建议性
 - ▶ 规范性
 - ▶ 标准性

- 标准规范的局限性
 - ▶ 大而全
 - ▶ 泛化，缺乏针对性
 - ▶ 偏重于技术文档规范



标准规范的使用（续）

■ 选择规范的原则

- ▶ 适用
- ▶ 适度
- ▶ 有效
- ▶ 可行
- ▶ 裁剪



文档的裁剪

- 文档类型的裁剪
- 文档内容的裁剪
- 裁剪并不是只剪不增



标准规范与文档管理

- 规范也是文档，也要纳入软件文档统一管理
- 文档也是配置项，要纳入软件配置统一管理



软件文档的质量要求

- 由于软件文档本身的优劣直接影响到沟通的效果，所以对软件文档的质量要求是相当高的。
- 一个高质量的软件文档可以有助于对软件的了解，而一个低质量的软件文档则可以会使阅读者难以理解、更加困惑。



软件文档的质量要求（Con.）

- 高质量的软件文档需具备：
 - 有针对性
 - 清晰性
 - 准确性
 - 完整性
 - 灵活性
 - 可跟踪性



文档编写人员的要求

- 有软件工程概念
- 有相关的技术经验
- 熟悉文档生成工具
- 了解文档管理流程
- 了解所需文档内容与要求



文档管理体系（例）

■ 通讯问题

▶ 电话分机号

▶ 通讯费用 C_n^2

▶ **N** 扩大，通讯费用呈指数级扩大

■ 文档

▶ 文档的产生

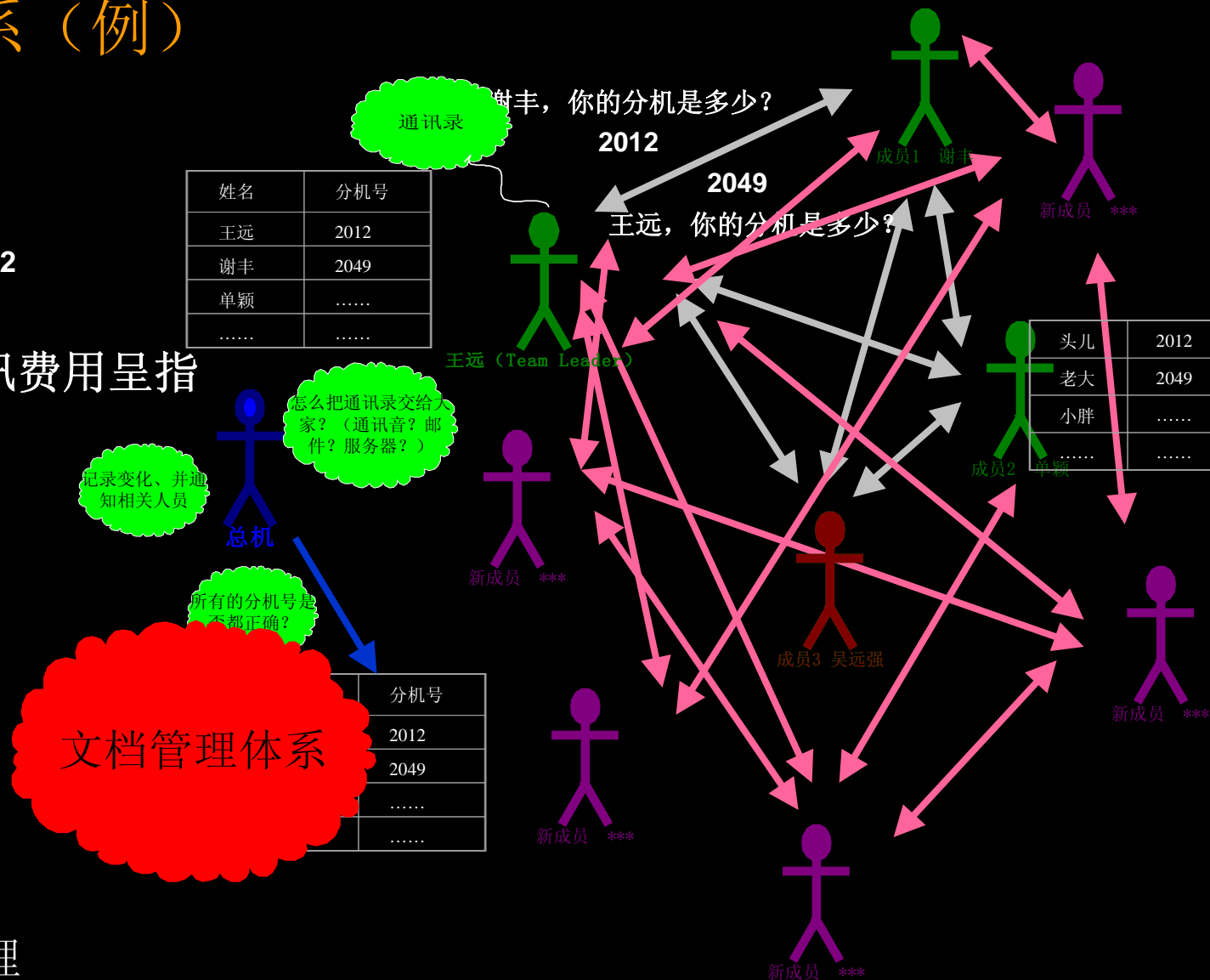
▶ 文档的格式

▶ 文档的书写

▶ 文档的评审

▶ 文档的发布

▶ 文档的变更管理



文档管理体系

- 文档类型
- 文档的批准和发布
- 文档的更改



软件文档的类型

按产生和使用的范围分：

- ▶ 管理文档
- ▶ 开发文档
- ▶ 用户文档

按软件生命周期分：

- ▶ 需求级别
- ▶ 设计级别
- ▶ 测试级别
- ▶ 验收级别
- ▶ 管理级别



软件文档的类型（续）

按适用范围分：

- ▶ 标准级文档
- ▶ 公司级文档
- ▶ 项目级文档

按重要程度分：

- ▶ 一级：需求性、设计性；
- ▶ 二级：源码性、测试性、验收性；
- ▶ 三级：计划性、度量性、流程性；
- ▶ 四级：报告性、会议纪要；



软件文档的类型（续）

软件文档类型划分的意义：

- ▶ 提供文档取舍的依据
- ▶ 提供制定文档生成时间次序的依据
- ▶ 提供文档评审等级的依据



文档管理体系

- 文档类型
- 文档的批准和发布
- 文档的更改



软件文档的批准与发布

- 获取文档规范
- 撰写
- 评审
- 发布



获取文档规范

■ 规范文档的命名

▶ 命名应能体现出项目、文档内容、时间性等要素

- XL-Project1-DOC-SRS-001
- 项目需求文档
- XL-Project2-STD-JAVA-001
- Java编程规范

■ 规范文档的使用环境

■ 提供统一格式模板

■ 提供必要的文档范例

■ 提供有效的获取途径

■ 更新的及时通报机制



获取文档规范（续）

- 模板设计目的
 - ▶ 帮助文档编写者理解文档标准；
 - ▶ 有利于统一文档编制格式；
 - ▶ 使文档编写者能结合具体项目，利用本模板方便、快捷地产生符合要求的文档。
 - ▶ 模板应对所使用的标准规范根据实际情况，已进行必要的裁剪。



获取文档规范（续）

■ 模板设计原则

- ▶ 模板应该充分考虑开发方法；
- ▶ 模板中的注释在适当的位置对文档应填写的内容和各表格的表头进行了说明；
- ▶ 为了使用户更好地理解如何编写具体文档，在相应的地方应插入实例，使用户可对照参考；



软件文档的批准与发布

- 获取文档规范
- 撰写
- 评审
- 发布



文档撰写

- 确定记录生存周期过程产生的信息所需的活动
- 目标
 - ▶ 找出由过程或项目产生的所有文档。确定所有文档的内容与用途，给出文档产生的计划与进度；
 - ▶ 文档开发应符合相关标准的规定；
 - ▶ 按确定的标准与指定的计划，开发出所有文档；
 - ▶ 文档应按照配置管理过程进行统一控制；
 - ▶ 保持文档间的一致性。



文档撰写（续）

- 文档名、文档版本的命名需要严格遵守相应规范；
命名应能体现出项目、文档内容、时间性、版本等要素
- 文档内容必须有且遵循统一的格式与形式；
格式与形式应尽可能以模板方式出现，以便于使用和统一。
- 文档中的文字描述要清晰、适当，没有歧义；
增加文档的有效性和可读性。
- 技术文档中尽可能不要出现抽象、不可量化的概念。
如美观、界面友好等。
- 纵向一致，不矛盾（即文档内部上下文前后一致）
任何变更都应在整个文档范围内统一
- 横向一致，不矛盾（即相关文档之间相互一致，任何变更都应在所有相关文档中统一）
需求变更—>设计变更—>测试变更—>用户手册变更



软件文档的批准与发布

- 获取文档规范
- 撰写
- 评审
- 发布



文档评审

■ 正规的评审流程

- ▶ 评审小组（所有相关人员参与）
- ▶ 评审会
- ▶ 评审结论
- ▶ 修改后再审

■ 非正规的评审流程

- ▶ 可根据实际情况由负责人员直接确定评审结论即可。但对于不合格的成果，仍需修改后再审。



文档评审（续）

- 文档评审的责任
 - ▶ 文档配置是否齐全
 - ▶ 文档编制是否符合规定的规范要求
 - ▶ 文档的内容是否符合要求。
 - ▶ 文档评审的关注内容，在生命周期的不同阶段有不同的侧重点



文档评审（续）

■ 文档评审的时间

- ▶ 通常是在开发过程中的每个里程碑阶段进行。
- ▶ 在某些重要文档需及时通报整个开发团队或其它协同工作人员时，也可紧急进行。

■ 文档评审的内容

- ▶ 各里程碑阶段的所有文档制品。
- ▶ 急需发布的所有文档制品。



软件文档的批准与发布

- 获取文档规范
- 撰写
- 评审
- 发布



文档发布

- 有效的权限控制
- 可访问性
- 稳定性
- 变更需及时更新与通报



文档管理

- 文档类型
- 文档的批准和发布
- 文档的更改



软件文档的变更管理

- 变更申请;
- 变更评审;
- 变更记录;
- 变更通知;
- 关联性变更的完整性。



文档变更

- 文档变更是难以避免的。
- 文档变更是可以控制的。
 - ▶ 文档变更请求的评审
 - ▶ 文档状态（草案、发布、冻结）
 - ▶ 文档的所有人
 - ▶ 文档变更后的评审、确认与发布
 - ▶ 文档变更后的一致性保持
 - ▶ 文档变更管理=配置管理中的配置项变更管理
- 冻结文档禁止变更。



文档变更（Con.）

- 纵向一致
- 横向一致
- 对横向相关文档进行变更统一
 - ▶ 若所有文档均未向外发布，变更应立刻统一，体现在相关文档中。
 - ▶ 若有文档已通过评审向外发布，则变更统一应在变更文件通过评审后，再与相关文档统一。



文档变更（续）

■ 文档的一致性

- ▶ 有效文档应该存在一个稳定有效的获取渠道。（如配置管理机、公司统一的文件服务器）。同时，要保证在此渠道上的真实有效性。（即维护的权限）
- ▶ 相关文档间的变更一致性，可以通过自动化工具（如ClearRequest, Doors）完成。在无工具情况下，应利用Word文档或Excel表格的方式，建立文档相关性矩阵或文档关联关系表，变更文档的发起人员有责任通过相关性矩阵或关联关系表，了解本文档变更的影响范围，并通知受影响人员完成一致性调整。



实施文档编制的常见效果

- 选取相应标准进行文档编制，在格式、内容方面取得了良好效果
- 模板大大节约了文档编制人员的时间与精力
- 个别的文档编制有可能不太理想
 - ▶ 通常测试文档问题会最大
 - ▶ 有文档代码不一致，最后补文档的现象
- 原因
 - ▶ 时间、人员、经费等因素的限制
 - ▶ 标准的通用性较强，而具体项目的可操作性不好确定。
 - ▶ 不够重视。



用户使用标准的希望

- 注重可操作性
- 给出标准的体系，大家各取所需
- 明确不同软件开发阶段所对应的标准规范
- 标准贯彻的支持（技术、管理、工具）



改善文档编写效果的方法

- 只选择编写对本公司有益的文档
- 使用有效的文档模板
- 使用有效的自动化工具
 - ▶ 如需求变更性工具，可以将需求的变更与设计、实施相关联。
 - ▶ 如设计工具，不仅可以完成设计工作，还能直接生成相应的标准文档。
 - ▶ 如缺陷跟踪性工具，可以将有效记录缺陷的修正过程。
- 有效的评审
- 有执的执行力度



文档体系的技巧

- 有效的规则
 - ▶ 命名规则等
- 有效的模板
- 有效的途径
- 有效的流程
- 有效的管理



建议考虑的文档

- 管理性文档
 - ▶ 开发流程
 - 配置管理流程
 - 测试流程
 - ▶ 评审流程
 - ▶ 开发环境
 - ▶ 编程规范
 - ▶ 文档管理规范
 - ▶ 报告制度（工作进度、周月报）



建议考虑的文档（续）

■ 技术性文档

▶ 需求文档

- 用户需求
 - 设计需求文档

▶ 设计文档

- 接口设计文档
- 系统设计文档
 - 架构设计文档
 - 概要设计文档
 - 详细设计文档

▶ 测试文档

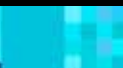
- 测试用例
- 测试报告
 - 测试计划

▶ 安装、使用文档



实例讨论

- 《开发流程规范》
 - ▶ 开发阶段划分
 - ▶ 各阶段人员的角色责任分配，
 - ▶ 各阶段的主要工作成果，即文档制品。
 - ▶ 配置流程
 - ▶ 测试流程
 - ▶ 评审流程
 - ▶ 书写、评审、发布等情况描述。



实例讨论

- 《需求文档》
 - ▶ 项目范围
 - ▶ 系统功能（ F e a t u r e ）
 - 需求编号
 - 需求描述
 - 图
 - 文字
 - 表格
 - 相关性问题描述
 - ▶ 书写、评审、发布情况描述。



关心您的软件工程实践！

UML软件工程组织

www.uml.org.cn

