



# 《软件工程与计算II》

## ch20 软件交付

南京大学软件学院



# 主要内容



- ➡ ■ 安装与部署
- 培训与文档支持
- 项目评价



# 软件交付



- 软件交付是软件项目的结束阶段，标志着软件开发任务的完成。
- 软件交付是软件开发与软件维护两个既连续又不同的软件产品生存状态的分水岭。
- 只有做好软件交付工作，才是真正地完成整个项目。



# 软件交付



## ■ 安装和部署

- 需求阶段：考虑环境约束等
- 体系结构设计阶段：进行产品部署的设计决策，包括网络拓扑、库文件、动态链接库、配置文件等
- 开发阶段：使用的支持软件也会影响到交付，可能要求客服安装特定支撑软件或者硬件



# 安装



- 安装是软件交付的最常见形式，现在大多数产品都通过安装的形式交付，它要求开发团队创建一个安装包，用户可以通过的执行将软件产品部署到工作环境。
- 安装包需要进行仔细的设计，并使用工具（例如 **Advanced Installer**、**Setup Factory** 等）帮助进行安装包的创建。一个好软件产品应该简单、健壮、可靠、完全。要创建很容易使用的安装包，让用户可以无需创建安装包的人员的帮助就能使用。



# 创建安装包的步骤



- 1 确定安装环境
- 2 列举安装清单
- 3 设计和建立安装包
- 4 测试安装包



# 1 确定安装环境



- 确定安装包需要支持的操作系统，这既需要考虑当前用户的工作环境，又需要考虑产品未来的市场推广；
  - 确定软件产品的语言支撑环境，例如使用**Java**语言开发的软件产品就需要安装**JDK**；
  - 确定软件产品需要的软件支持，例如数据库系统、网络系统等；
  - 确定硬件等其他要求，例如有些软件产品可能会要求扫描仪、视频卡、通信设备等特殊硬件。
- 
- 例如，对超市销售系统**MSCS**，它的安装环境为：**Window XP**、**Window Vista**、**Window 7**三种操作系统；**Java**运行环境**JDK**；数据库管理系统软件（如果使用了数据库的话）。



## 2 列举安装清单



- 要根据软件产品的实现情况，结合所需的环境支撑，列举需要安装的文件、初始化数据、注册表等清单信息，要清楚标明它们在安装后将会出现的位置。
- 在考虑安装位置时要遵守一致性，标记名称的使用要意义清楚，让用户能便利地找出相应文件。
- 例如，对超市销售系统**MSCS**，它所有的可执行程序文件都是需要安装的文件，初始化数据有两处，一处是设置默认的管理员用户帐号，另一处是设置数据库管理系统连接数据。





### 3 设计和建立安装包



- 要对安装包进行详细的设计，包括一个渐进的安装步骤，各步骤的人机交互方式等等。完成设计后就可以使用安装工具创建安装包。



- 例如，超市销售系统**MSCS**安装包可以按照下列步骤建立：
- (1) 检查操作系统环境；
- (2) 检查**JDK**，如果没有合适的**JDK**，则提醒用户安装**JDK**；
- (3) 检查数据库管理系统软件，如果合适的数据库管理系统软件，则提醒用户进行安装；
- (4) 设置数据库管理系统连接参数；
- (5) 连接数据库管理系统，创建**MSCS**的数据库；
- (6) 拷贝文件；
- (7) 设置初始化数据，包括数据库系统连接参数和**MSCS**的默认管理帐号；
- (8) 安装成功。



## 4 测试安装包



- 安装包需要在目标环境中进行安装测试，以发现可能的问题。
- 需要注意的是：必须以用户的工作环境为目标环境进行测试，因为用户使用的机器环境与开发者的机器环境有很大的不同（包括程序环境、操作系统版本、支撑软件版本等等），在开发者机器上可以正确执行的安装包未必能够在用户的机器上运行。



# 部署



- 在软件产品比较复杂时，仅仅通过一个安装包无法完成软件交付任务，这时可以使用另一种常见的软件交付方式——部署。
- 部署通常是由开发人员直接操纵软件产品的目标环境，使得软件产品能够在目标环境中正常运行。
- 部署的过程中通常需要执行安装任务，但是还有很多比安装复杂得多的其他任务，例如：安装、设置或调整操作系统，尤其是权限管理参数；安装、设置和调整数据库系统，包括新建数据库和设置访问权限；安装和设置库文件、应用服务器等应用环境。



# 部署的步骤



- 1. 确定部署环境
- 2. 确定部署任务
- 3. 完成部署准备
- 4. 执行部署任务



# 1. 确定部署环境



- 和安装一样，软件部署要首先要需要确定部署的目标环境，当然它比安装要求的更高一些。它需要对目标环境进行调查分析，搞清楚部署前的环境细节，然后才能与软件产品需要的环境细节进行比较，才能明确需要执行的部署任务。
- 具体来说，软件部署需要了解服务器与网络拓扑、安全控制与权限管理、软硬件系统的配置信息等。



## 2. 确定部署任务



- 将软件产品需要的目标环境与部署前的环境进行比较，分析二者之间的差距，并将其确立为部署的任务。
- 确定任务之后，还需要以渐进的方式安排任务之间的执行次序。例如，先安装和配置操作系统，然后安装和配置相应的软硬件系统，最后完成软件产品的安装与配置，等等。



### 3 完成部署准备



- 有些部署工作可以完全依靠现场执行，但多数的部署任务需要进行一定的事前准备，尤其是要综合考虑部署工作可能出现的各种情况，制定完备的应对方案。





## 4 执行部署任务



- 按照准备的计划，执行相应的部署任务。



# 主要内容



- 安装与部署
- ➡ ■ 培训与文档支持
- 项目评价



# 培训与文档支持



- 帮助用户理解产品，并使其能够轻松地使用产品
- 不能让用户学会使用软件产品，那么就不算是完成了软件交付任务
- 帮助用户学会使用软件产品的两个关键任务：
  - 培训
  - 文档支持



# 培训



- 培训主要是教会用户使用软件产品的功能来完成其工作和任务。依据任务的不同，要为不同的用户进行不同类型的培训。
- 例如，对超市销售系统**MSCS**
  - 培训收银员使用系统进行销售和退货
  - 培训客户经理使用系统进行库存管理和会员管理
  - 培训总经理使用系统制定销售策略和进行库存分析
  - 培训系统管理员进行用户管理。



# 培训的注意点



- 尤其不能忽略的是对系统管理员进行培训。要培训系统管理员如何启动和运行新系统、如何配置系统、如何授权或拒绝对系统的访问、如何支持用户、如何处理异常等。
- 在培训中，只介绍能够帮助用户完成主要工作和任务的功能，不要把培训当作软件产品所有功能的展示会。对于一些很少会被使用并且不太重要的功能，即使培训也会很快被用户忘记，可以让用户使用文档支持来学会使用。
- 培训时，要关注用户的工作和任务，不必涉及系统的内部操作，不必知道系统的存储方式、访问方式和权限控制方式。



# 文档支持



- 文档是软件交付的重要部分
  - 培训时作为参考材料
  - 交付完成之后继续帮助用户使用系统



# 文档支持



- 用户文档
- 系统管理员文档
- 简单的系统只有用户文档，绝大多数系统都有用户文档和系统管理员文档



# 用户文档



- 用户文档是指为用户编写参考指南或者操作教程，常见的如用户使用手册、联机帮助文档等，统称为用户文档。





# 用户文档的形式



- 用户文档可以是纸质的，也可以是电子的，可以只有一份文档，也可以是由多份文档组成的集合，具体情况要视用户的特点而定。
- 用户文档的写作要考虑用户群体的特点，最好是图文结合的方式，以方便普通用户的使用。用户文档写作应该使用逐层展开和系统化（例如层次编码、列表）的方式描述复杂内容。



# 用户文档的内容组织



- 文档内容的组织应当支持其使用模式，常见的是指导模式和参考模式两种。
- 指导模式根据用户的任务组织程序规程，相关的软件任务组织在相同的章节或主题。指导模式要先描述简单的、共性的任务，然后再以其为基础组织更加复杂的任务描述。
- 参考模式按照方便随机访问独立信息单元的方式组织内容。例如，按字母顺序排列软件的命令或错误消息列表。如果文档需要同时包含两种模式，就需要将其清楚地区分成不同的章节或主题，或者在同一个章节或主题内区分为不同的格式。



# 用户文档的要素



章节	是否必须
标识信息	是
目录	（正文超过8页时）是
图表目录	可选
引言	是
文档使用信息	是
操作模式	是
操作规程	是（指导模式）
软件命令信息	是（参考模式）
错误信息与问题解决	是
术语表	（文档中有陌生名词时）是
相关信息源	可选
导航特征	是
索引	（文档正文超过40页时）是
搜索能力	（电子文档中）是



# 用户文档要素



## ■ 标识信息

- 包括文档标题、文档产生的版本和日期、相关的软件产品和版本
- 标识信息应该放在包装袋或者封面，用户可以不用翻阅文档就能看到

## ■ 引言

- 正文的第一部分，描述文档的预期读者、描述范围，以及对文档目的、功能和操作环境的概要描述



# 用户文档要素



## ■ 操作规程

- 指导模式文档应该包括很多软件功能都会涉及的常见活动规程：
  - 需要由用户执行的软件安装与卸载
  - 图形用户界面特性的使用指导
  - 访问、登录或者关闭软件
  - 通过软件的导航，访问和退出相关功能
  - 数据操作（输入、保存、读取、打印、更新和删除）
  - 取消、中断和重启操作的方法



# 用户文档要素



## ■ 操作规程

- 对于完成用户任务的操作流程，指导模式文档应该从基本信息、指导步骤和结束信息三个方面来描述
- 基本信息：
  - 简要概述操作规程的目的，定义或解释必要的概念
  - 标明执行任务前需要完成的技术活动
  - 列举用户完成任务所需要的资源情况，例如数据、文档、密码等



# 用户文档要素



## ■ 操作规程

- 对于完成用户任务的操作流程，指导模式文档应该从基本信息、指导步骤和结束信息三个方面来描述
- 指导步骤：使用祈使语句描述用户行为，并指出预期的结果。指导步骤要说明：
  - 用户输入数据的域值范围、最大长度和格式
  - 相应的错误消息和恢复办法
  - 其它可选的步骤和重复步骤



# 用户文档要素



## ■ 操作规程

- 对于完成用户任务的操作流程，指导模式文档应该从基本信息、指导步骤和结束信息三个方面来描述
- 结束信息：标明操作规程的最后步骤，让用户知道怎样判断整个操作规程的成功完成，告诉用户如何退出操作规程





# 用户文档要素



## ■ 软件命令信息

- 解释用户输入命令的格式和操作规程，包括必要参数、可选参数、缺省值等，要示例说明命令的使用，说明怎样判断命令是成功完成还是异常中止

## ■ 错误信息与问题解决

- 文档要详细描述软件使用中的已知问题，让用户清楚如何自行解决问题或者怎样向技术支持人员报告准确的信息

## ■ 导航特征

- 包括章节、主题、页码、链接、图标等



# 系统管理员文档



- 与用户文档注重系统使用细节不同，系统管理员文档更注重系统维护方面的内容，例如系统性能调整、访问权限控制、常见故障解决等等。因此，系统管理员文档需要详细介绍软硬件的配置方式、网络连接方式、安全验证与访问授权方法、备份与容灾方法、部件替换方法等等。



# 主要内容



- 安装与部署
- 培训与文档支持
- ➔ ■ 项目评价



# 为什么要进行项目评价



- 设置“项目”是要保证项目中的各种事件与活动能够依照计划顺利进行，项目评价就是检查其事件与活动的实际执行情况。在理论上，项目评价可以发生在项目进行的任何时机，尤其是到达各个里程碑之后。但最重要的项目评价是在项目结束时进行的项目评价，这也是本章节所要描的项目评价。
- 虽然从单个项目看，项目已经结束，评价似乎用处不大。但是考虑到一个组织会有很多项目持续进行，那么评价一个已结束项目就可以“以史为鉴”，帮助更好地完成后续项目。而且因为项目已经完成，总结和评价就远比项目进行中更加准确。
- 项目评价工作也需要仔细组织，不是简单的开个总结会完事，否则就无法获得比较深入的信息。



# 项目评价的内容



- 一个已结束的项目具有各种事件和活动的信息，通过组织对项目的不同方面内容进行评估，就可以获得各种不同方面的经验，就可以搞清楚出现了哪些问题、为什么会出现、怎样解决、有哪些偏差、最终结果与质量、以及（最重要的是）在下一个项目中有哪些需要提高。
- 常见的项目评价针对四个方面：
  - 项目管理：可以帮助建立对项目的更准确认知，例如常见的管理问题与偏差、时间与成本耗费分布等。
  - 产品：可以帮助开发者建立对产品的更准确认知，提高产品的开发经验。
  - 团队：可以帮助开发者更好地组织分工，也可以帮助团队建立更好的沟通与交流途径。
  - 个人：可以帮助开发者更准确认知自己的生产力，学习常见问题及其处理方法，了解自己的长处和不足并持续提高。



# 项目评价方法



- 评审
- 度量数据分析



# 评审



- 项目评审通过评审重要项目制品的方法来评价项目，这些重要制品包括项目计划、管理文档、会议记录、历史数据等。
- 成功的评审需要评审方法，而不是自由处理。检查列表是最为常用的评审方法。



### 有关项目管理的问题:

- 项目所使用的过程是什么?
- 实际的过程与原先确定的过程有什么不同?
- 进度表是如何随着时间的变化而改变的?
- 有多少个同步点和里程碑按时达到或错失?
- 过程的哪些部分运行得好?
- 过程的哪些部分本应该能运行得更好?
- 工具支持这个过程吗?
- 从整体上讲, 这个过程运行得有效吗?
- 在今后, 尤其要对哪些方面进行改进?
- 在每个阶段和每项任务上花费的时间是多少?

### 有关产品的问题:

- 在项目的生命周期中, 产品是如何变化的?
- 有没有出现重要的产品返工的情况? 如果有, 是在什么时候?
- 工具支持产品的制造、维护和测量吗?
- 产品最后的规模有多大?
- 产品的质量如何?

### 有关团队和个人的问题:

- 团队(个人)工作中哪些个风险发生了, 其影响又是怎样的?
- 在何时做出了哪项重要决定?
- 所遇到的主要问题是什么?
- 这个决定又是如何影响这个项目的?
- 对这些问题的解决方法产生了什么样的效果?
- 开发团队成员是如何看待自己的职责的?







# 度量数据分析



- 度量数据可以提供丰富的信息，通过分析这些信息，开发团队可以获取正确和深入的结论。
- 例如，通过分析项目活动的任务量，就可以了解每个人的生产力、项目的工作量分布、特殊任务的工作量耗费等。



# 产品信息定量的度量



- 一个项目常见的产品信息度量应该包括：
  - （随着时间而变化的）产品的增长情况和变化历史。
  - 产品在每个里程碑上的测量。
  - 产品复杂度和内容的测量。
  - 过程和工具对产品的影响。



# 定性文件



- 在进行度量数据分析时可能会遇到数据贫乏——这意味着没有足够的定量数据来支持项目评价，这时可以用问卷调查表和面谈来补充数据信息。也可以通过检查定性文件来建立数据信息，这些定性文件可能包括：
  - 对团队会议和子团队会议所做的记录。
  - 项目电子邮件的存档（来获得问题确定和决策的日期）。
  - 任务列表、项目决策和行动条目中的信息。



# 评价的注意事项



- 一、项目的评价需要仔细的计划
- 作为项目管理活动的一部分，项目评价也需要进行计划，计划的内容包括：
  - 执行项目评价的时间，要在项目结束后，并且不能时间太久导致项目活动细节遗忘；
  - 确定项目评价的关键主题；
  - 确定参与项目评价的人员；
  - 确定需要收集的数据，并将数据收集任务分配给相关人员。
- 二、评价要客观
- 对项目的评价要客观，要保持对项目和过程的关注，不要偏离目标指责和突出个人。如果不能做到客观，列举没有进行分析的测量数据或信息，而仅仅为了表明整个项目是一个巨大成功的话，那就无法得到有益的经验，就是浪费时间。评价不是向高级管理层夸夸其谈的文档，而是团队每个成员和组织通过一个又一个项目来不断获得提高的途径。



# 总结



- 不要忽视软件交付阶段的任务
  - 通过安装与部署将软件产品移交给用户
  - 通过培训与文档支持保障用户能够有效掌握和使用软件
- 一个项目的成功或失败都值得总结，以改进将来的项目，即要在项目结束后及时进行项目评价