

# 第7章 软件运维与服务过程

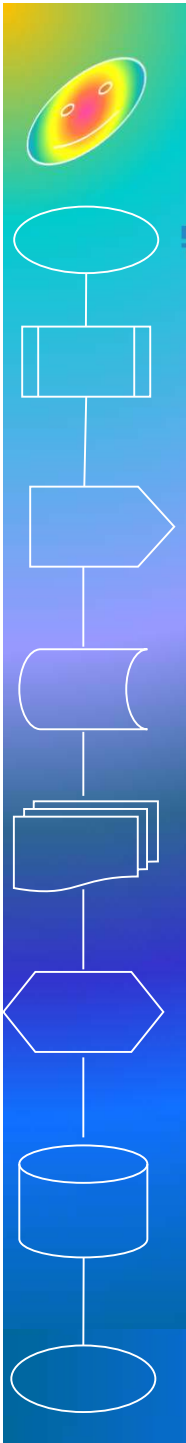
软件是服务，需要运行与维护

# 目录

- 7.1 引言
- 7.2 软件移交与验收过程
- 7.3 软件支持过程
- 7.4 软件支持过程与活动
- 7.5 软件支持方式与能力
- 7.6 独立的软件支持
- 7.7 总结

## 7.1 引言

- 建立软件的目的是满足使用的质量和可信赖性要求，从而给用户和客户带来明显的经济效益或相应的社会利益。
- 软件不仅是可运行的程序，也包括了软件系统所处理的数据和信息，运行该软件的规程和执行规则，以及支持或维护软件正常运行的各种活动。
  - 软件的运行和支持运行的维护过程简称为运维 (Operation and Maintains) 过程。

- 
- 维护软件的费用可能是软件开发费用的2~100倍，具体随应用领域和系统的规模而定。
  - 软件系统使用的时间越长，维护费用可能越高。
  - 许多运维问题是由于软件支持和服务工作中，供应商和客户方的争执所引起的。

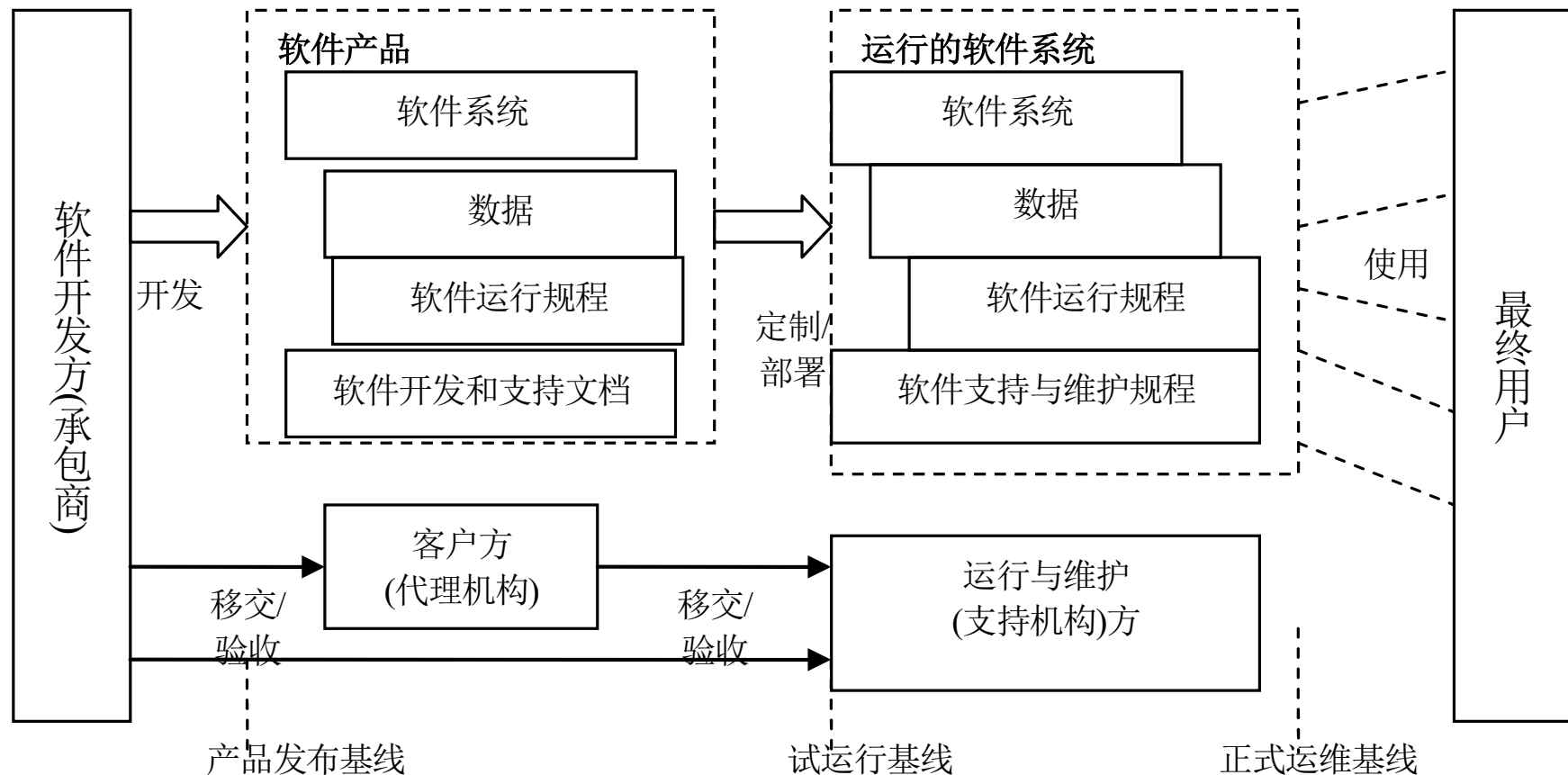
# 服务的范围

- 包括：

- 1) 解释软件的使用；
- 2) 编写软件手册和其它文档；
- 3) 在软件使用过程中，帮助和指导客户的使用；
- 4) 针对硬件给出建议；
- 5) 修补引起宕机、引起故障的缺陷；
- 6) 维护和修正崩溃的数据；
- 7) 处理由于软件修改所引起的缺陷；
- 8) 清理由于软件误用(未按规定的使用或设计的方式使用)所引起的错误；
- 9) 清理操作系统或设备的错误；
- 10) 持续地培训软件用户；
- 11) 修改软件；
- 12) 清理缺陷(包括保质期之后的)，等等。

## 7.2 软件移交与验收过程

软件验收和移交是软件采购过程(见图6-2)的逆向过程。客户方把此过程称为验收过程，开发方称为移交过程。



## 7.3 软件支持过程

- 7.3.1 软件维护与传统产品维护
- 7.3.2 Lehman定律
- 7.3.3 维护成本

## 7.3.1 软件维护与传统产品维护

- 在传统工程中，维护是指修理和更换已经损坏或磨损的部件。
  - 然而软件不会损坏、更不会磨损。软件维护与传统工程维护有所不同，有时需要“重新开发”。
- IEEE把软件维护定义为：在提交后，为纠错、改进性能或其它属性、或者适应环境变更要求，对软件系统或部件进行修改的过程。
- 软件维护研究表明，用户需求变化对系统部署后支持费用的影响占41%，硬件变化所产生的维护费用占10%左右。
  - 也就是说，一半以上的费用是系统外部环境的变化所引起的。



# 软件服务的有形和无形比较

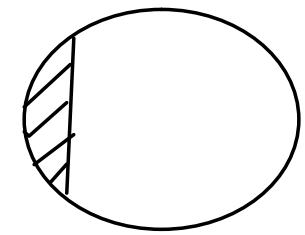
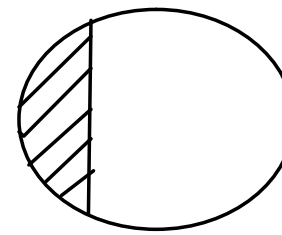
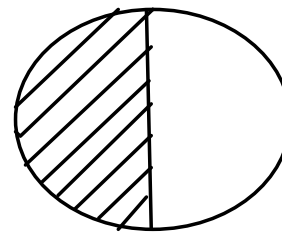
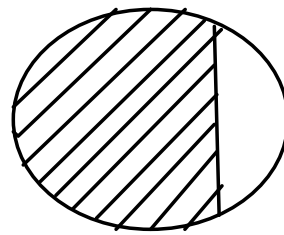
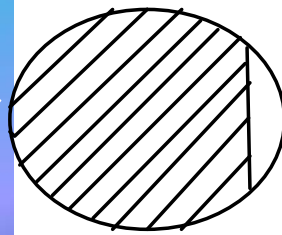
几乎纯产品

产品密集

产品与服务混合

服务密集

几乎纯服务



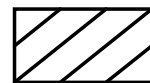
塑料包装的  
软件，例如，  
Windows 操  
作系统

客户化  
软件开发

完善性的  
维护服务

纠错性的  
维护服务

几乎/完全服  
务，例如，搜  
索引擎等



有形部分



无形部分

## 7.3.2 Lehman定律

定律	含义
不断变更律(1974 年)	计算机程序必须被更改, 否则在此环境下就会越来越没用处。
复杂性不断增加律 (1974 年)	随着程序的更改, 程序的结构越来越复杂。除非增加工作量简化其结构。
自身调节律(1974 年)	系统进化过程具有自身调节的规律, 产品和过程的测量结果接近于正态分布。
组织稳定(不变工作频度) 守恒律(1978 年)	在整个产品生命期内, 开发和维护等活动的频度近似于一个常数, 与系统开发所花费的资源无关。
熟悉程度守恒律 (1978 年)	随着 E 型程序(见 4.1.3 节)的进化, 开发者、销售人员、用户等, 也必须保持对程序的内容和行为的熟悉程度, 以获得满意的进化。极度增长(或快速变更)将消弱其各方对程序的熟悉程度。因此, 随着系统的进化, 增量的增加要保持均匀不变。
不断增长律(1991 年)	E 型系统(见 4.1.3 节)的功能必须不断增加, 以维护用户对其的满意度。
质量下滑律(1996 年)	系统的质量将出现下滑, 除非得到认真的维护, 并且能适应运行环境的改变。
反馈系统律(1974 年提出, 1996 年形成)	E 型系统(见 4.1.3 节)的进化过程是多层面、多循环的反馈系统, 必须在合理的基础上进行改进。

### 7.3.3 维护成本

- 统计显示，需求阶段或者定义和理解更改的需求占整个成本的15%；理解已有的系统，包括评审文档和设计(或重新设计)占30%；更改的实现占25%；产品或系统的检验(checkout)，包括测试与调试，以及改写文档占35%的成本。
- 支持软件运行的活动成本，特别是支持软件密集型系统运行的成本已经成为影响软件整个生命周期成本的重要因素。**软件危机不仅仅是开发阶段的危机，更重要的是软件运行和维护阶段的危机。**
- 美国国防部门和工业界的研究表明，典型的维护成本占整个软件全生命周期的60%~80%。用户方和开发方均面临着如何降低软件的维护成本问题。

# 软件修改类型

- 支持活动的重要环节是对软件系统进行修改。软件修改按其目的一般分为：
  - 1) 纠错性(Corrective)---其目的是消除系统中的缺陷;
  - 2) 适应性(Adaptive)---其目的是修改软件项, 让其能够满足环境的变化;
  - 3) 完美性(Perfective)---修改软件项, 满足已有能力的基础上, 改进其风格;
  - 4) 增强性(Enhancement)----为系统增加更多的功能, 或提高其性能。

# 开发时，要考虑如何维护

- 对软件维护考虑的不足必然增加软件维护费用，一度导致软件生产效率的大幅度降低，软件维护活动导致软件的生产率下降了40倍(按每个人月的代码行计算)。
- 这样数据是重要的证据，要求软件开发者必须在需求和设计阶段考虑维护问题，从系统的可理解性、可修改性、互操作性、可重用性、可扩展性和可移植性等角度提高系统可维护性。

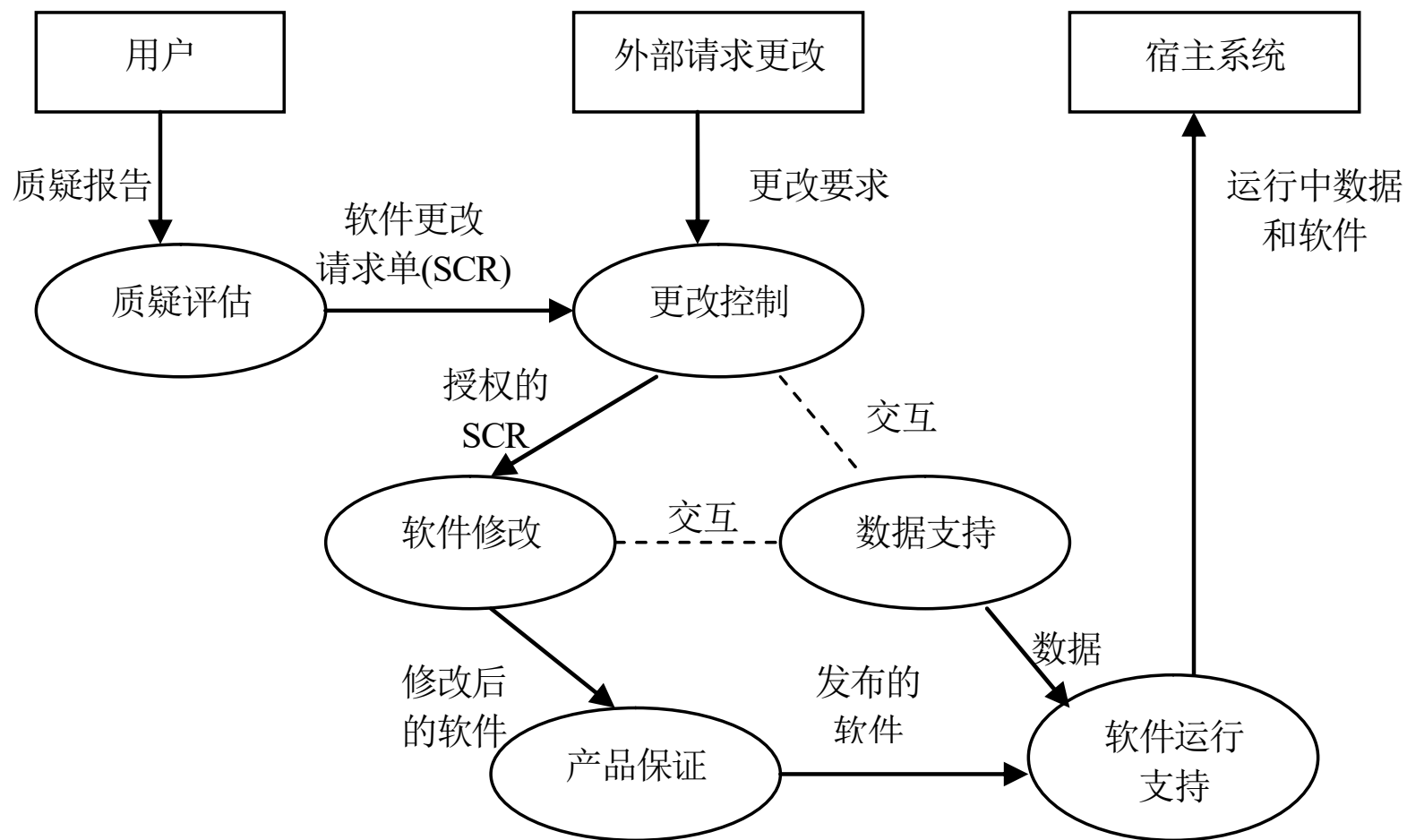
## 7.4 软件支持过程与活动

- 7.4.1 软件修改过程
- 7.4.2 软件支持活动

## 7.4.1 软件修改过程

- 对运行中的软件系统进行修改的主要推动因素是：
  - 内部要求系统更改的需求
    - 用户(系统运行和支持人员)在使用中提出和发现问题，以及对系统的适应性和增强提出一些想法。
    - 这些都可以收集为系统的“质疑单”，从而形成质疑报告。
  - 外部请求更改并不来源于用户的意见，在不改变系统功能的情况下，更改软件接口部件或者基本硬件配置。

# 软件更改的过程





# 修改请求单(SCR)的问题

- 1) 修改请求单(SCR)是随机产生的, 用户不太可能在年度财政预算中完全估算出来SCR个数;
- 2) 依据系统运行等级影响, 对SCR进行优先权排序, 而不是由高层管理者决定;
- 3) 维护工作量不能按项目管理技术进行管理, 而要用依据紧急程度进行排队管理;
- 4) 细分SCR, 让每个SCR的规模和复杂程度可以用一个或两个资源(适当的人力和工作量)就可处理, 即, 处理SCR的工作量是确定和可控的;
- 5) 维护工作量要按“面向服务用户”和“面向应用责任”的方式计算;
- 6) 可以随时间改变SCR的优先权, 对系统的修改请求可以依据其它工作进展决定其优先权。

# 是否真的要改？

- 为了平衡上述的情况和因素，最好设立一个更改控制小组(或委员会)，对SCR进行授权和排序，以满足系统运行能力和稳定的要求。
- 更改控制小组处理用户提出的SCR以及外部请求更改单。
- 更改控制小组要依据更改的目标、约束条件、可使用的资源等，协调更改请求。
- 最好对更改请求进行可行性和影响性分析。
- 由于运行的软件和数据是互相依赖的，因此，不仅要意识到软件代码的更改还要意识到数据更改对系统产生的影响

# 修改活动

- 软件修改
  - 实施软件修改活动，依据授权的SCR对软件进行修改。其活动的输出是新的可以在宿主(host)系统上运行软件和支持文档。
- 数据的修改：
  - 由于软件与数据是相互依赖的，在更改软件的同时，必须同步进行数据的修改、备份、转储和恢复等，以保证系统在修改过程无缝地转入正常运行。
  - 数据支持工作主要包括从宿主系统中相关的转储数据和恢复数据，以及配置发布的软件。数据支持工作包括数据的创立、保存、分析和修改，其工作结果是新装入的数据和使用文档。

# 修改后产品质量的保证

- 产品保证的目的是验证和确认修改的产品是否可以发布和运行，在完成了授权的SCR修改后，系统是否仍然达到所期望的可靠性、安全性、密安性等可信赖等级的要求。
  - 重新发布的软件被正确地配置并有相应的文档只是最低要求。
  - 最低要求只能提供质量要求的完整性证据，而不能表示产品是完整的。
- 要达到完整性，维护工作的每一步都必须进行质量保证活动。

## 7.4.2 软件支持活动

- 软件的开发过程看做为“前向工程”，那么，软件的维护是一个“逆向工程”过程。
- 由此产生了“逆向工程”、前向工程、重新结构化、以及重新编写文档的概念：

## 7.4.2 软件支持活动

- **逆向工程(Reverse engineering):**

- 是对已有软件进行检查分析，并抽象出其设计和基本需求的过程。即，充分分析和理解已有的系统以便能够对其进行更改。

- **前向工程(Forward engineering):**

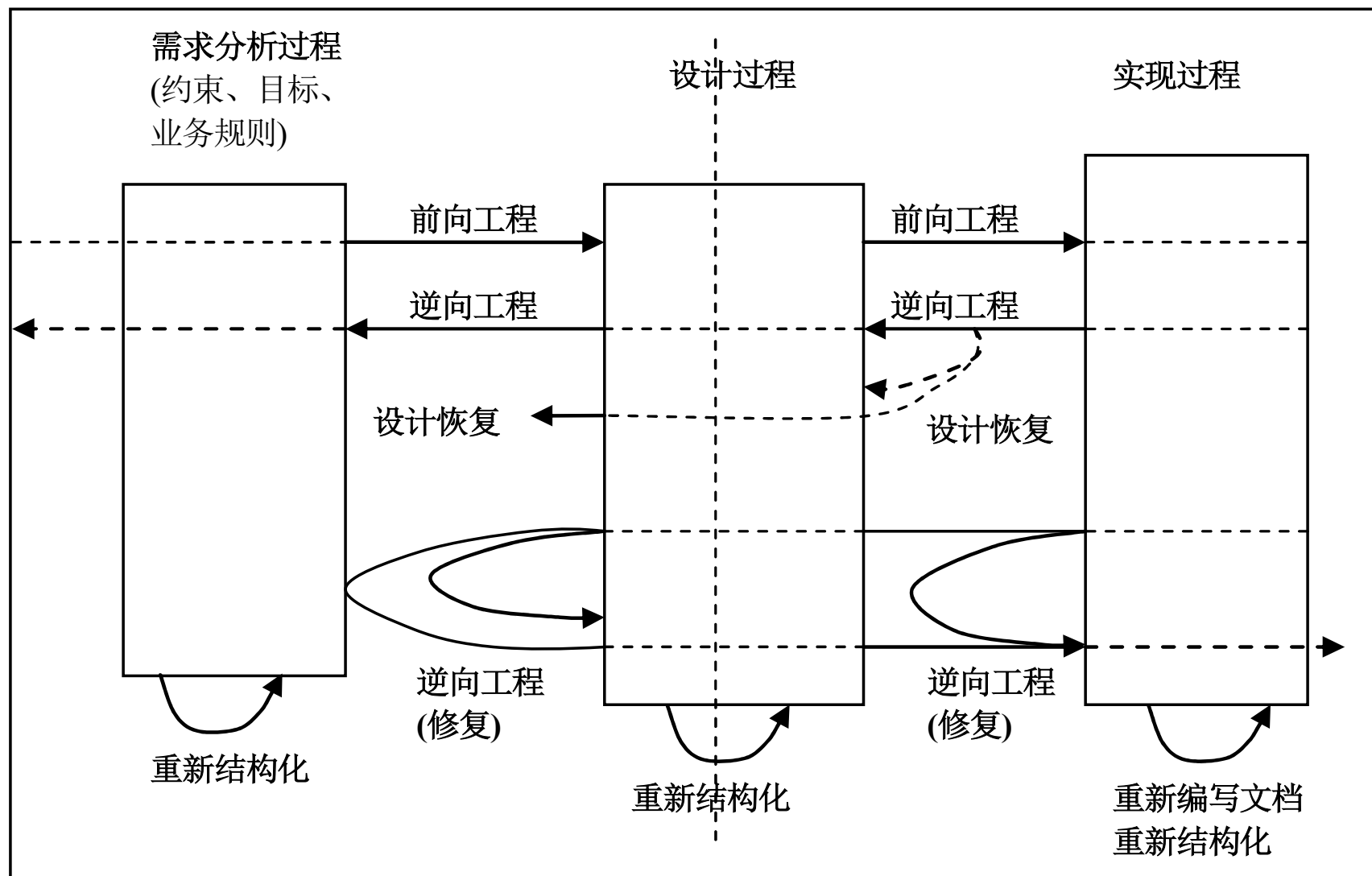
- 是使用现有产品或从已有的系统中派生出来的制品，以及新需求，生产出新系统的工程过程活动。

- **重新结构化(Restructuring) :**

- 在保持软件外部特征不变的前提下，将已存在的系统重新组织，或者将已有的系统从一种表达形式转换成另一种表达形式的过程。重新结构化的目的是理清系统的结构，以便降低系统的维护和修改成本。

- **重新编写文档:**

- 其目的是分析已有的软件，产生各种支持系统运行的文档，包括用户使用手册，以及重新梳理源代码，形成格式化的源代码，提高其可读性。



## 7.5 软件支持方式与能力

- 7.5.1 软件支持方式与基本要求
- 7.5.2 针对软件可使用性的支持
- 7.5.3 针对运行能力的支持



## 7.5.1 软件支持方式与基本要求

- 软件支持工作可能有客户方(系统的使用方)为主体实施的，这种情况比较常见。在中国，很多软件使用方都会建立单独的信息中心或网络中心，承担支持本单位的软件系统运行和一般性的维护工作。
- 支持过程也可能是有独立的支持机构完成，例如，一个军队的作战软件系统。使用该系统的是一个战斗编制部门（如一个作战师），而软件的支持机构可能是后方的文职技术人员，而不是软件的原始开发方，也不是具体的战斗编制部门(用户)。

# 服务或支持条款

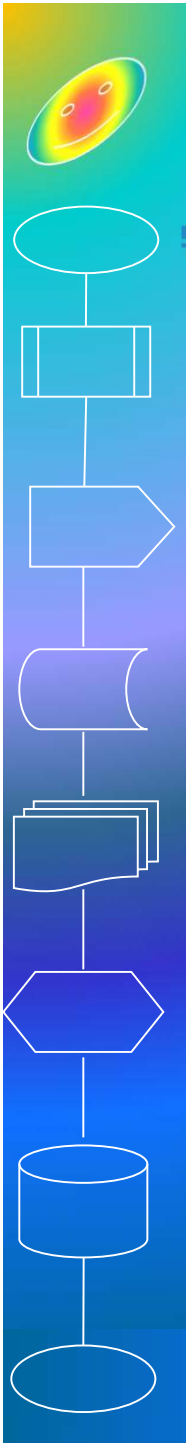
- (1)对纠错性修改：一般会写入双方的合同中。如果执行纠错修改的费用超出供货商的费用，就需要补充一个软件保质单。保质单可以遵循如下的形式：
  - (a) 终身保质单（Lifetime Warranty）。这种保质条款要求纠错活动在软件生命周期内是终身保修的。
  - (b) 有限期保质单(Limited Life Warranty)。在规定的时间内，所做的纠错活动是免费的。一旦过了保质期，客户需要单独交付纠错活动的费用。
  - (c) 无保质单(No Warranty)。客户需要交付所有纠错活动的费用。

## (2) 对于适应性和增强性修改

- 其目的是满足系统的支持能力。与纠错修改不一样，一般客户都需要付出费用。软件开发方一般会通过发布新版本的方式要求客户方支付(全新)的费用，同时会给老客户一定的价格优惠。
- 为得到这种优惠和持续的支持服务，在订立软件合同时，最好要在一个后设计服务(PDS—Post Design Services)合同或条款中提出对后续版本升级提出具体的要求。
- 当客户方和供应商签订支持合同时，双方必须考虑相关的问题，并在合同中用文字规定各自的责任和问题处理方式。供应方至少要考虑：
  - 1) 使用何种硬件平台？2) 客户具备哪些能力？3) 哪些已知技术可以用于软件支持？4) 客户对保密安全有哪些要求5) 软件对客户的业务影响又多大？6) 客户如何维护整个系统方案的完整性？

## 7.5.2 针对软件可使用性的支持

- 对于许多“任务关键或安全关键”的系统，例如，军事、银行、高速铁路的等软件系统，一般的支持和服务模式往往不能满足客户对系统的安全性、可持续的使用等方面的支持要求。
- 因此，要定义出“可使用性合同(CfA---Contracting for Availability)”

- 
- 英国国防部提出在需要软件的适应性和增强性修改时，必须建立PDS合同：
    - (1) 对于纠错性修改，修改前必须测量和评价系统的可使用性，包括所有降低可使用性的故障属性。CfA的定义完全依赖于对软件的可使用性定义，因此可以替代软件保质单的要求，引导软件开发方建立和执行更有效率的软件支持方案。
    - (2) 对于适应性和增强性修改。由于这种修改处于系统层面上。仅靠CfA不能满足软件修改的要求。因此要增加合同中的能力支持要求。

### 7.5.3 针对运行能力的支持

- 综合上述的针对软件可使用性、系统和信息保密性等的支持，以及其它运行特性的支持，可归结为对系统能力的支持要求。英国国防部门要求在CfA之外，建立能力合同（CfC----Contracting for Capability）。
- CfC的支持方案中要求：当系统与预先定义要求发生矛盾时，必须体现出对软件能力的支持，而不能仅仅限定合同范围。因为人们很难在软件采购合同中，设想清楚所有可能发生的问題。

# 软件支持合同类型

- 总之，
  - 一般的软件支持仅仅需满足通常的修改要求。
  - 而软件CfA要求针对系统规格说明的修改建立相应的可使用性变更的准则。
  - 软件CfC则要求软件修改可以满足系统运行需要的不断进化(软件修改和升级)的要求。

## 7.6 独立的软件支持

- 7.6.1 软件支持的原则
- 7.6.2 系统部署前后的支持工作
- 7.6.3 采购对支持工作要求



## 7.6.1 软件支持的原则

- 美军方为了更有效地支持其软件的运行和维护工作，在MIL-HDBK-347标准规定：
- 基本的原则：
  - 要求软件支持活动必须贯穿于整系统的采购过程。即，从使用方(特别是军方)的系统采购开始，就要考虑和落实未来的支持活动。
  - 软件的进化是软件开发周期的后续活动。从软件开发的第一个版本开始，一直在后续版本的软件开发周期中，都要持续地支持；
  - 任何软件都需要持续进化，只有通过对软件修改才能实现，才能满足软件的基本需求，以及合理的费效比。

# 独立的支持机构实施软件支持

- 原则：

- 软件支持活动必须由特定的支持机构(例如，军队的后勤支持部门)负责向采购方的项目经理提出部署后的支持要求。并从费效比的角度出发，要求支持工作随着软件系统的一起移交。
- 对关键计算机资源，要有特定的机构进行支持，以保证支持活动可被审计，并提供快速响应、可控制，连续的和兼容的服务。同时，这些特定机构要识别出军方或政府可控制的软件基线，定义和评估软件质量需求，并进行系统级的测试，以及管理和策划部署后的支持工作。
- 最终用户团体(例如，军队的作战单位)必须能够与独立的支持机构直接交流，识别和隔离软件问题，并合作解决支持问题，而不需要与软件原始开发方交流。

## 7.6.2 系统部署前后的支持工作

- 软件部署前后的软件支持活动  
(第一个软件版本开发和移交)

支持工作阶段	软件支持活动	
部署前的软件支持 (第一个软件版本开发和移交)	定义部署后的支持工作	<ul style="list-style-type: none"><li>●给出部署后软件支持的策略</li><li>●调查可备选的软件支持概念</li><li>●定义“任务关键的计算机资源产品”</li></ul>
	提出支持需求和计划	<ul style="list-style-type: none"><li>●选择部署后软件支持概念</li><li>●提出软件质量需求</li><li>●识别软件支持的资源需求</li><li>●开发计算机资源生命周期管理计划</li><li>●开发移交计划</li></ul>
	确认软件能够得到支持	<ul style="list-style-type: none"><li>●确保软件的可支持性</li><li>●评估软件质量</li><li>●认证软件文档和技术数据</li><li>●维护计算机资源生命周期管理计划</li></ul>

## (后续软件版本的开发和进化)

支持工作阶段	软件支持活动	
系统部署后和运行中的支持  (后续软件版本的开发和进化)	在移交过程中, 实施可支持	<ul style="list-style-type: none"><li>● 实施移交计划</li><li>● 采购和安装软件工程环境和软件支持环境</li><li>● 采购软件支持活动所需的资源</li><li>● 进行人员培训</li><li>● 演示部署后的软件支持能力</li></ul>
	支持工作的常 态化	<ul style="list-style-type: none"><li>● 管理好部署后的软件支持工作</li><li>● 执行部署后的软件支持工作</li><li>● 提供软件产品后勤保障</li><li>● 维护计算机资源生命周期管理计划</li><li>● 评估和维护软件质量</li><li>● 管理好软件配置</li></ul>

## 7.6.3 采购对支持工作要求

- 在采购过程中，客户方的项目经理角色是至关重要的。
- 除正常的项目采购外，必须采购支持系统，以满足支持未来系统运行的支持团队的工作要求。包括：
  - 定义系统支持概念，经费预算和进度计划，需求确认，规格说明的验证，质量保证，配置管理，准备采购计划、编写工作条款和合同要求，对软件进行独立验证和确认，保密要求，测试和评估，系统支持和后勤分析和策划等工作。
- 客户方的项目经理必须明确地建立和提出足够的资金支持以及综合的后勤支持程序。为软件开发合同提供全面的需求，包括软件支持活动的资源需求、采购必要的资源以及最终提供支持的需求等，而不仅仅是可运行软件本身的需求。

# 软件系统采购过程的支持活动要求

软件采购过程		(独立的))综合后勤支持活动要求
系统采购各阶段	对支持的要求	描述
概念研究/定义	定义支持	●定义综合后勤支持策略; ●分派软件支持活动 ●调查可备选的的支持概念; ●定义出相关产品
概念演示/确认	设计支持方案	●识别/定义/评估每个主要系统分的替代方案的保障措施 ●主要系统替代方案的支配选择
大规模开发	设计支持项目	●完成后勤支持体系的设计 ●保证综合后勤支持是设计折中方案的一个完整部分
规模生产和开始部署	采购支持项	●采购所有必需的支持项
运行	提供支持	●提供系统后勤支持



## 7.7 总结

- 软件运行和维护是保证软件充分发挥能力，保证运行质量和可信赖性的关键因素之一，会比开发阶段花费更多的费用和工作量。
- 软件维护和服务支持是软件生命周期重要组成部分。
- 只有不断地对软件维护、支持和服务，才能够提高软件的使用质量，扩大市场。
- 大多时候，很难把软件维护与开发绝对的分割开来。
- 在软件采购和开发阶段，就须考虑软件系统的可维护性和可支持性，并建立与软件系统质量和可信赖性要求相适应的支持与服务过程和方法。

# Homework

- 读书上p.553的案例，或按如下地址下载：
- <https://www.cs.kent.ac.uk/teaching/09/modules/CO/8/86/rdl/CaseStudies/Reports/reportLondonAmbulanceS.pdf>
- 站在甲方的角度，分析项目失败的原因。