

第6章 软件销售与采购

软件是商品，是经济活动

目录

- 6.1 软件类型
- 6.2 软件产品销售模式
- 6.3 软件项目采购模式
- 6.4 OTS的采购和使用
- 6.5 总结

6.1 软件类型

- 6.1.1 现货软件
- 6.1.2 可复用软件
- 6.1.3 按合同开发的软件
- 6.1.4 开源软件
- 6.1.4 软件类型与质量

6.1.1 现货软件

- 现货软件(**OTS--Off-The-Shelf**):
 - (a) 商业现货软件(COTS --Commercial-Off-The-Shelf software)
 - (b) 政府现货软件(GOTS--Government-Off-The-Shelf software)
 - (c) 工业现货软件(IOTS--Industry-Off-The-Shelf) 。
 - (d) 可修改的现货软件(MOTS--Modified-Off-The-Shelf)

6.1.2 可复用软件

- 可复用软件(**Reused software**): 由企业内部开发的软件, 被用于不同的项目。
 - 复用软件将先前写好的功能直接纳入到新项目中, 需要知道在新项目中如何使用。
 - 由于在系统A中运行良好的软件, 并不意味着你能在系统B中也运行良好。因此, 必须进行可适应性分析。

6.1.3 按合同开发的软件

- 合同软件(**Contracted software**)或客户软件(**Custom Software**):
 - 是采购单位或用户委托承包商或子承包商开发的软件。
 - 项目在定义的软件必须满足的需求的同时, 还要定义出开发过程要求和安全等要求。

6.1.4 开源软件

- 开源码软件(**Open Source Software**):
 - 与通常的只提供软件的机器代码的销售方式相反, 开源码软件的提供者同时提供源代码和一些平台上的机器代码。
 - 开源软件可能是政府支助开发的, 也可能是企业主动公开源代码的软件。
 - 许多开源码软件缺乏相关的文档, 更没有有有关软件中的可靠安全、信息安全等质量方面的分析和善意提示。这会给开源码的使用者带来诸多的风险。
 - 从商业模式上, 使用开源代码开发出的系统, 也要开放其新开发的源代码。

6.1.4 软件类型与质量

| 软件类型 质量评价指标 | COTS | MOTS | 完全新开发的软件 | 开源码软件 |
|---------------------------|------|--------------|-------------------|------------------|
| 功能范围 | 固定 | 部分可定制 | 客户安全可定制 | 可修改和重用 |
| 是否适合于使用 | 演示 | 采用类似系统 演示 | 事先不可知 | 可演示 |
| 易维护 | 不可控 | 部分可控 | 完全可控 | 部分可控 |
| 交付时间 | 立即交付 | 较短时间 | 取决于项目， 可能会很长时间 | 取决于开源中的 缺陷情况 |
| 采购费用 | 低到中 | 中到高 | 较高 | 可免费用，但需要 公开修改 |
| 质量(ISO9126) (见第 4.4 节) | 不可控 | 部分可控 | 大部分可控 | 部分可控 |
| 可信赖性 (见第 5 章) | 不可知 | 部分不可知 | 能够做到可知 | 需要花时间确定 |

6.2 软件产品销售模式

- 6.2.1 软件产品销售行为
- 6.2.2 最终用户协议
- 6.2.3 点击同意协议
- 6.2.4 二次开发的皇税协议
- 6.2.5 开源协议
- 6.2.6 GNU、BSD、MIT协议
- 6.2.7 免许可证软件
- 6.2.8 国际间的软件版权

6.2.1 软件产品销售行为

- 由于软件的可复制性，用户极可能购买一份软件，而安装到多台机器上使用。
 - 针对这种行为，自然需要从法律角度约束软件的使用权----软件许可证(license)。
- 许可证是知识产权(IPR-- Intellectual Property Rights)的表达，表明软件产品的使用权益和责任。
- 软件颁发者在许可证中描述软件的使用限制，例如，使用的时间段、地域等，以及相关的免责声明，避免由于软件错误给使用者造成损失时的法律责任。

法典的漏洞

- “美国法典”第17章，USC117，允许计算机程序的拥有者可以有必要备份。通常情况下，专有软件许可证将用简单的语言解释117条款。

- 例如: "You may use the software on one computer, and you may make an additional copy to be used only for backup or archival purposes. You may not otherwise copy, modify [...] the software." (“你可以使用一台计算机上的软件，你可能会作出额外的副本仅用于备份或存档目的。您不能够以其他方式复制，修改[...]软件。”)

- 然而，许多时候人们会利用USC117的漏洞，例如“拥有者(Owner)”的副本和仅仅是“拥有(possesses)”。

- 在Internet网络上，可以通过链接和引导等形式使用其他人的软件和信息，模糊软件拥有者和软件引导者的概念。

6.2.2 最终用户协议

- 最终用户协议是许可证提供者(Licenser)和购买者之间契约，规定了购买者使用软件的权利。
- 用户不接受此协议，就无法安装软件。
 - 例如，一般情况下，一台预装Microsoft Windows的电脑，开始使用前必须接受EULA才能启用系统，根据Windows EULA的规定，用户可以选择联系供应商要求退货。

6.2.3 点击同意协议

- 点击同意协议(clickwrap agreement或clickthrough agreement 或 clickwrap license)是互联网世界中常有的软件协议。
 - 其主要出现在软件包的安装过程中。这个销售行为是“薄膜包装合同(shrink wrap contracts)”的扩展，一但拆封就意味着用户同意该软件规定的商业条款。
 - 在安装过程中，软件用户一旦点击同意协议后，就不能再反悔。

Zeidenberg与Pro CD公司的案例

- 1996年，一名研究生Matthew Zeidenberg购买了ProCD公司制作的电话目录数据库软件(SelectPhone)光盘，其中有3000个电话目录。SelectPhone软件的研发成本超过一千万美元。为收回此成本，ProCD区分了商业用户和非商用的普通用户两个版本。
- Zeidenberg购买的SelectPhone是普通用户版本。他打开包装安装到PC机上，并创立了一个网页，将该软件中的原始信息提供给网站的访问者，收取比ProCD公司要低的费用。
- 在购买时，Zeidenberg没有意识到软件的使用限制，但是其包装中有相关的许可证，并在安装过程中在屏幕上出现过。Zeidenberg没有花时间阅读许可证条款。直接点击就安装了此软件。
- 法院认为软件许可证是合法的且是可行的。主要依据是UCC(Uniform Commercial Code)的第2-204条款(描述合法合同)和2-206条款(描述合同的接收条件)。
- 法院要求Zeidenberg接受“点击通过(clicking through)”协议，并认为：软件在屏幕上闪现了许可证条款，如果没有确定接受，他可以不继续下一步。法院指出，Zeidenberg拒绝了合同条款，就要返还软件。法院断定：依据UCC法案，Zeidenberg不能按商业版本的条款使用该软件，并允许退还该软件。

6.2.4 二次开发的皇税协议

- 而次开发商购买一些代码库(或中间件)的软件,并将其嵌入或集成到卖给最终用户的软件或设备中。
- 针对这类情况,所集成的软件代码的原始供货商会采用Royalty(皇税或版权费)模式,基本计算公式为:

设备的软件成本 = 工具价格 + 版税 × 生产的设备个数

6.2.5 开源协议

- 比较普遍接受(或者说, 比较规范的)开源代码许可证是Open Source Initiative (OSI), 其基础是开源定义(OSD --Open Source Definition).
 - 其给出了10条要求(<http://opensource.org/osd.html>)

6.2.6 GNU、BSD、MIT协议

- **GNU General Public License (GPL) 许可证。**
 - GPL在保证所有开发者的权利的同时，为用户提供提供了足够的复制，分发，修改的权利。可自由复制，可以将软件复制到你的电脑、客户的电脑，或者任何地方。复制份数没有任何限制。
- **BSD(Berkeley Software Distribution)开源协议限制比别的开源协议（如GNU GPL）要少。**
 - 该协议有多种版本，最主要的版本有两个：新BSD协议与简单BSD 协议。这两种协议经过修正后都与GPL 兼容，并为开源组织所认可。
- **MIT 协议是几大开源协议中最宽松的一个，核心条款是：**
 - 软件及其相关文档对所有人免费，可以任意处置，包括使用、复制、修改、合并、发表、分发、再授权、或销售。唯一的限制是，软件中必须包含上述版权和许可提示，包含许可声明。

6.2.7 免许可证软件

- 免许可证软件(License-free software)是一种宣称了版权，但不提供许可证的软件。例如， Daniel J. Bernstein提供的qmail、daemontools和ucspi-tcp。Bernstein开始宣布了软件版权，并分发其产品，2007年12月28日将其放到公开网站上，并告诉大家可以随便使用其软件(<http://cr.yp.to/distributors.html>)。
- 许多小的脚本程序的发布也是没有许可证的。或者，很难规定其权利和限制条件。从用户的权利角度看，Bernstein认为在版权法律仪式上，软件是允许被修改的，如果你只是为了自己更好用，可以不管许可证如何规定。

6.2.8 国际间的软件版权

- 法律条文：
 - 美国法典”第17章(USC117)作为计算机版权和许可证制定的法律依据。
 - 欧洲议会于2009年通过的计算机程序保护法(on the legal protection of computer programs)
 - 中国发布的《计算机软件保护条例》
- 国家之间也会因为对软件知识产权的理解问题而发生纠纷。特别是发达国家对发展中国家在软件版权的保护方面给予过多的指责。
 - 如果一味地要求用户遵循软件所有者的许可证条款要求，很难完全地解决国际之间的软件知识产权纠纷。为解决这些问题，有些软件公司针对不同的国家和地区发布不同的软件版本。

6.3 软件项目采购模式

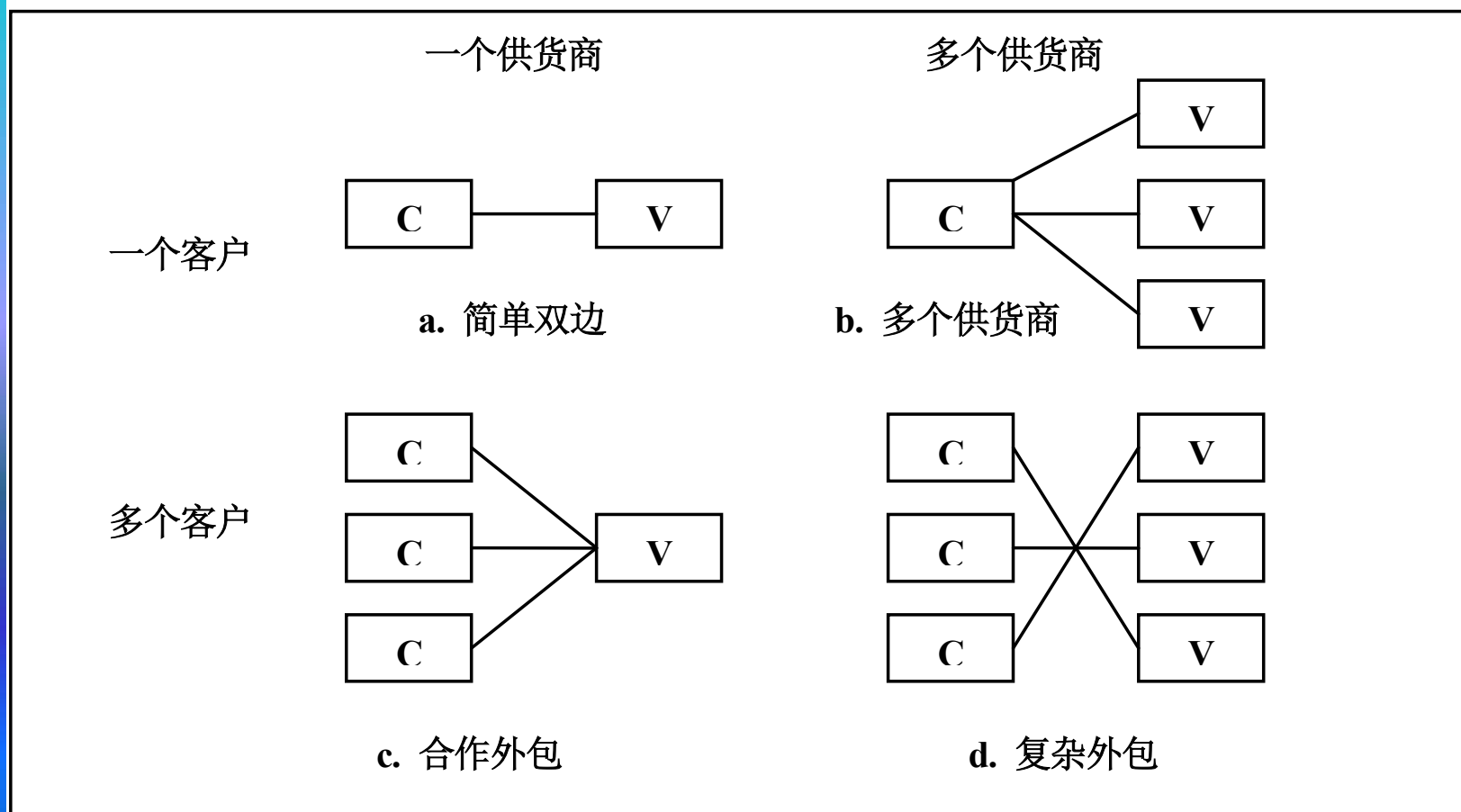
- 6.3.1 软件采购和外包形式
- 6.3.2 软件采购过程
- 6.3.3 软件采购主要问题

6.3.1 软件采购和外包形式

- 在软件市场上，大量的软件交易是以项目形式开发方式进行的。由此形成了最终用户(enduser)、客户(customer)和承包商(contractor)等不同角色，他们对软件项目的理解和目标的要求会有很大差别。

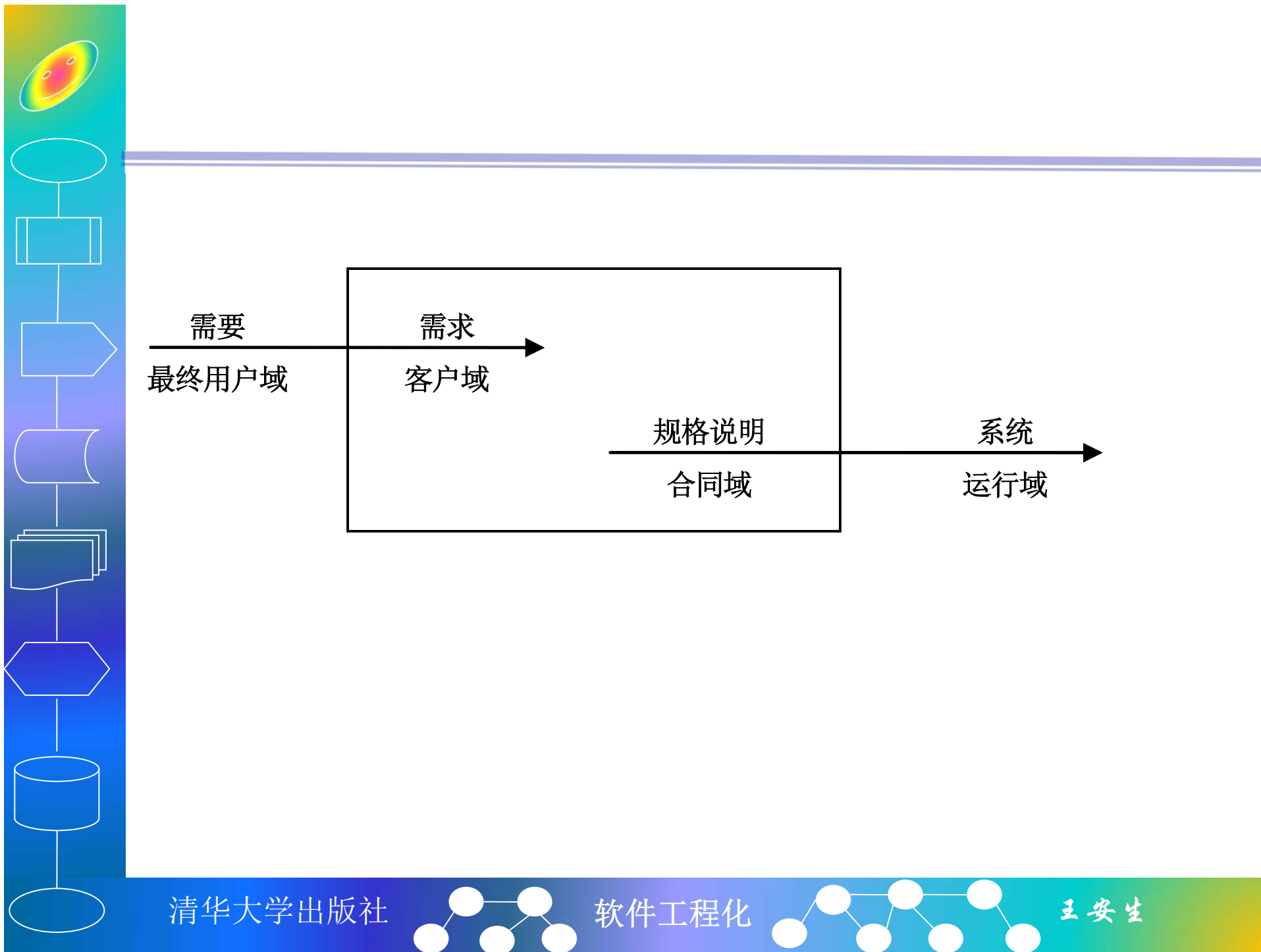
| 群体 | 期望的目标 |
|---------|--|
| 最终用户 | 期望的是软件产品或系统的功能能很好的运行，以及软件项目能尽快投入使用。 |
| 客户(甲方) | 关心几个目标：1) 及时获得能很好的运行软件能；2) 项目费用最少；3) 项目时间计划的偏差最小；以及，4) 能得到很好的投资汇报；5) 处理好与承包商之间的关系。 |
| 承包商(乙方) | 期望获得最大的利润，并且期望今后能获得更多的合同。 |

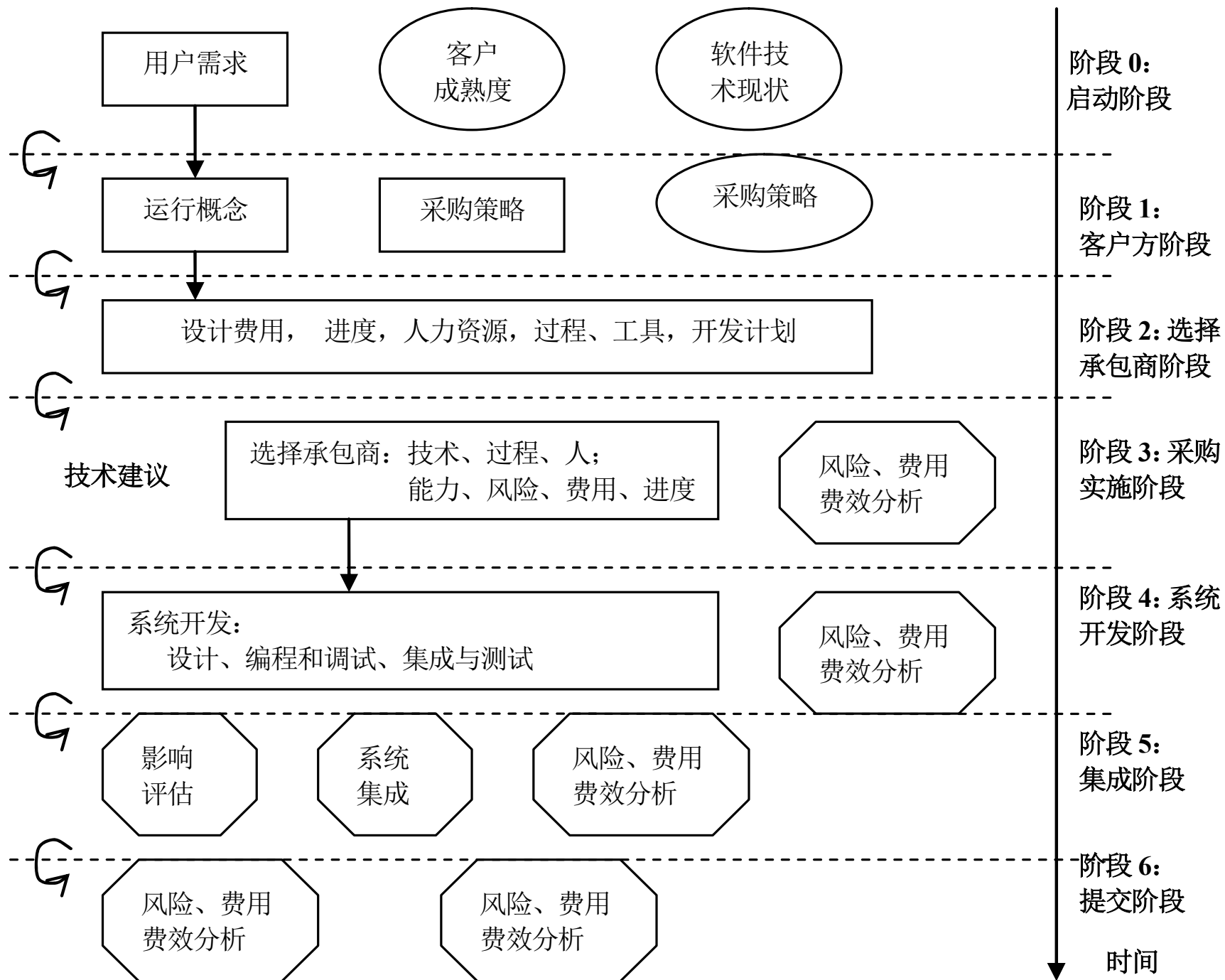
软件外包关系的分类



6.3.2 软件采购过程

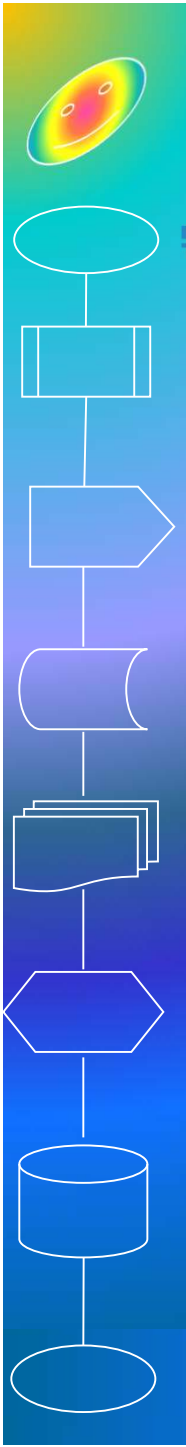
- IEEE-1602a建议针对项目开发软件和商业现货软件(MOTS)采购分为9个步骤。
 - 1) 策划组织的采购策略。评审采购者的目的, 开发出进行软件采购的策略;
 - 2) 实施组织的软件采购。建立一个适合于组织需要的软件采购过程;
 - 3) 定义软件需求。定义需要采购的软件, 准备质量和维护计划;
 - 4) 识别潜在的供货商;
 - 5) 准备合同要求;
 - 6) 方案评审, 并选择供应商;
 - 7) 监督和管理供应商的行为和性能;
 - 8) 软件验收;
 - 9) 软件运行。



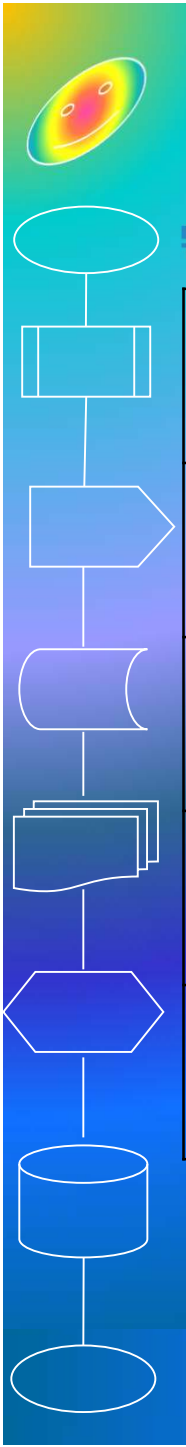


6.3.3 软件采购主要问题

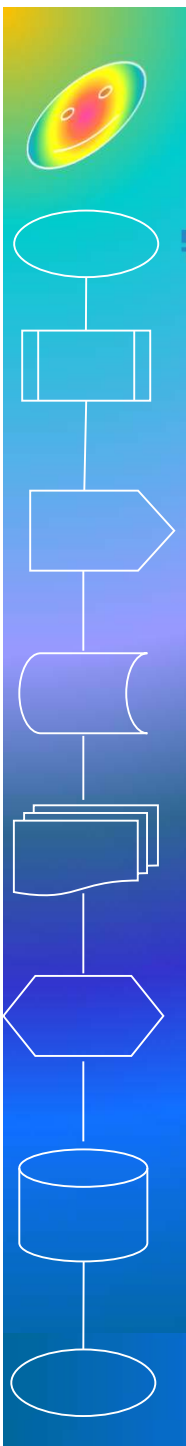
| 主要问题 | 主要责任方 |
|--|---------|
| 1) 自由放任(laissez-faire)----合同签订后，客户方并没有积极的管理活动，放任了对承包商的过程跟踪和监督。 | 客户方 |
| 2) 过度的行政管理性工作 (administrative overload) ---太多的精力放在监督合同上，而忽视了技术方面的管理。 | 客户方，承包商 |
| 3) 范围蔓延(scope creep) ----在项目进行中，时间和资源不够时，客户方坚持增加和更改软件范围和功能，引起需求的量变到质变。 | 客户方 |
| 4) 支离破碎(fragmentation)----客户方和承包商的项目团队成员随机被拉入到其它项目中。双方项目组不能实施原先的目标。 | 客户方，承包商 |
| 5) 种金子(Gold plating)----提出过分的需求，或做出复杂的和全新的解决方案，而不是使用简单和已证明的技术方案。 | 客户方，承包商 |



| | |
|--|--------------|
| 6) 我已付钱了(I'm paid to engineer!)----客户方会告诉承包商如何工作，而不是说要需要完成哪些工作。客户方甚至认为自己比承包商能力更强。 | 最终用户、客户方 |
| 7) 错误的指示器(Missing indicators)---对项目进展和整体性能的测量是定性的，而缺乏定量的度量。指示器给出的分级太粗略，无法准确判读。 | 承包商、客户方 |
| 8)谁负责(Who is charge ?) ----项目中有太多的老板和领导，不能及时作出决策。 | 最终用户、承包商、客户方 |
| 9) 缺乏最终用户参与(No end user involvement)---不能从最终用户的角度看待软件产品的功能性和可使用性。最终用户往往不懂需求，客户方一定要了解最终用户的需求，或组织承包商从最终用户角度捕获需求。 | 客户方、最终用户 |
| 10) 定义的需求太差(Poorly defined requirements) ---客户方与承包商之间的合同条款是不完整的和不可验证。合同中缺乏技术条款论述，或技术条款不可验证。 | 客户方、承包商的 |



| | |
|---|---------|
| 11) 采购缺乏竞争力(Acquisition incompetence)----没有理解软件采购的特定需求。 | 客户方 |
| 12) 夸大承诺(Overpromising) ----承包商的市场人员所承诺的是技术人员无法做到的。 | 承包商 |
| 13) 工程过程缺乏纪律(Lack of discipline)---迫于项目期限的压力，编写代码成为主要的工作。 | 承包商 |
| 14) 不现实的期望(Unrealistic expectations) ----不可能的进度，没有意识到技术的限制。 | 客户方、承包商 |
| 15) 缺乏充足的资源(Inadequate resources)---没有合适的经费，缺乏合适的员工、工具和设备。 | 承包商 |



| | |
|--|---------|
| 16) 缺乏执行支持(No executive support)----从顶层管理上缺乏对项目的支持。 | 客户方、承包商 |
| 17) 没有清晰的目标(Lack of clear objectives)---没有清晰地陈述目标和预期，从而导致项目成员在方向上出现分歧。 | 客户方、承包商 |
| 18) 沟通不够(Ineffective communications)---有效的交流渠道，信息不能及时地传达给合适的人。 | 客户方、承包商 |
| 19) 缺乏竞争(Lack of competence)----没有合适的技术和领导技能。 | 客户方、承包商 |
| 20) 争执(Friction) –某些原因导致相关方的合作不顺畅。 | 客户方、承包商 |



6.4 OTS的采购和使用

- 6.4.1 一般问题
- 6.4.2 安全性考虑
- 6.3.3 密安性考虑

6.4.1 一般问题

- 为了表示采用OTS软件的风险和可信赖性，诞生了专用叫法：血统不确定软件(**SOUP---Software of Uncertain Pedigree**)或源产地不清楚软件 (**Software Of Unknown Provenance**)。SOUP强调了OTS来源不清所带来的潜在问题和可能的缺陷。
 - OTS可能是由完全遵循软件工程实践和标准的开发队伍开发的，也可能是由一些夜以继日，吃着汉堡，喝着饮料的程序员匆匆拼凑出来的。
 - 或许是具有特定目的开发者免费提供的，或许是敌对势力和竞争对手有意提供的软件。
 - 敌对势力植入逻辑炸弹，并在适当的时候引爆，从而破坏整个软件系统。
 - 商业竞争对手可以将版权信息等植入在软件中，等待用户在使用中破坏版权行为的出现，通过法律手段要求远远高于采购该软件价格索赔。

6.4.1一般问题(续)

- 在软件项目开发中，出于费用和进度的考虑，OTS软件已经成为各个行业的公共部件。如果项目经理们能够找到现成的OTS或先前项目开发好的部件在新项目中被复用，就可以比自己开发节约时间和费用。
- 但是采购和采用OTS会具有极大的风险和安全问题。
- 从其他项目中复用过来的软件，即使是非常类似的项目，也不可能是完全一样的。系统的细微差别都将导致破坏性的结果。
 - 欧洲的Ariane 5火箭首次飞行程序复用了Ariane 4的软件，确忽略了两者在倒计时阶段的差异，从而成为导致Ariane 5首次飞行爆炸的因素之一。

采购OTS时必须考虑的问题

- 1) 是否需要用粘连件将OTS集成到待开发的系统中?
- 2) 如何扩展粘连件?
- 3) 当OTS不能提供全部功能时, 是否必须通过粘连件增加功能?
- 4) OTS软件是否有多余物(功能), 待开发的系统是否需要对这些多余功能进行保护?
- 5) 为了验证OTS质量和可信赖性, 还需要哪些分析工作?
- 6) 必须增加哪些额外的测试或认证工作?

6.4.2 安全性考虑

- 像医疗和核工业应用这样的“安全关键”系统中采用OTS时，也必须专门讨论COTS软件的使用问题。
- 美国NASA的约翰逊空间中心（JSC ---Lyndon B. Johnson Space Center）的工程指导分部认为航天安全的工作基础与FDA的“医疗器件中采用OTS软件指导”是一致的，并以EA-WI-018文件的形式指出了含有OTS软件项目的生命周期，包括安全性考虑。

OTS的问题

- 1) 文档不充分：往往只有用户手册，只是从用户观点描述其功能。
- 2) 缺乏源代码。阻碍对系统的安全性分析。阻碍对实际软件如何工作的理解。
- 3) 缺乏该软件如何创立的开发过程的了解。
- 4) 缺乏对软件进行验证的测试过程的了解。
- 5) OTS的开发者并不能完全理解系统元素之间的交互关系，也不可能与购买者完全交流信息。
- 6) 对OTS的缺陷细节、以及已知的错误，也不能完全提供该购买或者。
- 7) 对软件的分析不够。
- 8) 遗漏功能。OTS软件提供主要的、而不是全部的需要功能。遗漏的部分必须用粘连件弥补。
- 9) 多余的功能。OTS软件往往会包含一些不必要的功能。有时，这些功能是“关闭的”，除非OTS软件被重新编译，否则，这些代码仍保留在系统中。

6.3.3 密安性考虑

- 与实物部件被盗窃对比，信息泄露或被偷窃后往往并不能被立即发现。
 - 特别是涉及到国家、军队、商业运行安全的高度机密和绝密信息，敌对方或商业竞争者不会轻易使用所盗取到的信息。
- 敌对方会选择在足以对国家、军队或商业机构构成足够的威胁或造成不可弥补的损失时，才会使用盗取的信息。

OTS的密安性考虑

- 1) 密安性与系统体系结构，以及与其他系统的接口密切相关；
- 2) 软件密安性贯穿与于整个设计阶段；
- 3) OTS的密安性是采购经理报告的重要组成部分，并必须把OTS密安性要求和评价情况传递给项目经理；
- 4) 特定的编码规则和要求是预防信息泄露和代码被攻击的重要措施；客户方和承包商必须就密安性要求，规定程序的编码规则，防止代码的漏洞。参见第12.5节。
- 5) 在代码实现阶段，通过编码规则，增强密安性；

OTS的密安性考虑

- 6) 系统集成中，同样把密安性作为重要的要求，对集成过程进行验证；
- 7) 测试计划和规程要保证软件系统能完成其功能，而不能超越功能之外；
- 8) 软件的部署和安装也要能够保护密安性要求。
- 9) 通过管理更改过程，保护系统和OTS部件的密安性；
- 10) 系统必须在保证密安性的方式下运行和维护；
- 11) 最后，在系统退出现役时，要将系统分解和废弃，而不能丢失或泄漏敏感数据或信息。

- 采购OTS的目的是能够快速生产、部署和使用可信任的、可预测的、以及能与系统的密安策略相一致的软件系统。
 - 可信任意味着进行了非常严肃的工作，将系统中那些偶然的、有企图的、可被利用的脆弱点降到了最低点。
 - 可预测意味着系统功能准确和可靠地完成所需的意图具有极高的概率。
 - 一致性意味着“计划的和系统的多学科活动保证了软件过程和产品符合需求和应用标准和规程”。

6.5 总结

- 好的软件是买卖出来的”
 - 软件产品的良好销售决定了用户，也成就了软件在使用过程中不断的改进，软件质量就会越来越好。
 - 对软件产品和软件项目的采购方式和过程，决定了软件的需求和承包商的选择，从而导致了软件质量、工期、成本、乃至未来的使用。
 - 软件采购和项目外包必须考虑质量和可信赖性问题，才能把握整个系统的可信任程度。
 - 在市场经济下，“销售和采购”能力是推动软件工程化的重要因素之。

Homework

- 假设你代表我们学校，打算委托给某个公司给我们学校做一个学生信息管理系统。你作为学院的代表（甲方经理）
- 分析你在采购（包括，选择承包商），跟踪软件项目过程，并验收该项目中可能遇到的风险和对策
- 目标：
 - 1) 没有贪污腐败现象
 - 2) 项目成本在可控的范围内
 - 3) 工期短（可以多次迭代发布）
 - 4) 上线后，系统能稳定运行
 - 5) 不要被学生、教师和领导吐槽