在 GAMP 监管行业中应用开放源代码软件（OSS）的指导

ISPE《制药工程》： May/June 2010, Vol 30 No. 3

作者： Markus Kaufmann, Maucus Ciolkowski, Andreas Hengstberger, Till Jostes, Erwin Kruschitz, Thomas Makait, Karl-Heinz Menges, Stefan Munch, 和 Martin Soto

翻译：杨彬，艾默生过程控制有限公司

审核：张林忠，厦门特宝生物工程股份有限公司

介绍

开放源代码软件（OSS）和商业软件（COTS）的区别比许多业主想象的少得多。本文章重点描述了免费的开放源代码软件相对于商业软件的好处，以及其在 GxP 合规领域中的确认和操作方法。

商业软件一般是由公司按照传统方式开发和市场化，通常需要支付版权费用，软件代码是不能开放的并且用户没有权利修改软件。

免费的 OSS 是那种用户拥有某些权限或自由度的软件。权限大小取决于选择的软件许可，通常有修改源代码的路径并能自由的重新发布。

由于 这种软件的开放特性，开放源代码软件应用的一个主要动机在于可以减少商业风险；系统的所有细节能够被详细地确认和审核。从另一方面讲，现今的验证方法需要强调以下几个方面：

- 团队不再是 传统的商业伙伴。沟通方式和工作方法可以是不同的，例如，服务等级的协议不再适用或者团队可能由于某种原因不愿和你一起工作
- 传统的供应商审计对 OSS 团队无法实施，必须使用其它的供应商评估方法。

免费的 OSS 已是现今软件行业的一个重要组成部分，已经存在许多重要的商业应用，并呈持续上升的势头。因而，多数安装包含 OSS 软件的系统，在一定程度上成为“基础软件”的一部分（按照 GAMP 5）。他们通常部分或全部使用 LINUX 和 MYSQL 数据库组成 OSS(OSS 的应用)，尤其对于这些基础软件，他们节约了大量的经济成本。

国际监管机构通常没有区分商业软件和 OSS，因此，公司可以自由地从二者选择其一。所有 GxP 监管的计算机系统在使用前必须验证，以确保系统适合于特定的用途。对于数据完整性的控制措施和验证范围的决策，必须是基于文件化和可评价的风险评估。对患者安全和产品质量的影响以及数据的完整性和适用性，必须在过程中进行评估。清晰的接受标准应基于风险评估设定并且记录在验证方案中。

开放源代码软件的特性

对于 OSS，一个通常的错误概念是其由一些无私的个人专门开发的，因而，它应该总是免费

的。其实，无论 OSS 的开发还是规划都包括了许多重要的投资企图，我们有理由不能忽略这点。了解 OSS 的商业和许可模型、开发过程、和支持结构，是其能否适合成为商业的决策基础，相关方包括 OSS 团队、商业软件公司、系统集成商、咨询顾问、用户/健康行业和监管单位。

商业模型

“这些支持的开发者是如何赚钱的？”就像其它的软件开发项目，一个 OSS 项目所需的成本与特定产品的规模和复杂性直接相关。在商业软件的开发中，工作由开发者带薪完成的，由用户通过直接的开发合同或其它的许可模式支付财务资源。

在 OSS 项目中，开发的付出主要来自于：

- 个体免费使用部分的个人时间为项目工作。

  公司或其它组织直接或间接承担为项目工作的编程人员或其它专业人员的费用。在第一种情况中，个体的动机是非常广泛的，从具有利他动力（例如，OSS 有助于战胜贫困和不平等）的技术挑战，到试图通过在有影响力项目中提供有价值的贡献建立他们个人的声望。从另一方面讲，公司和一些自由的专业人员，对于围绕他们的 OSS 开发新的商业机会有相当的兴趣。许多的商业模型可能是：

  - 推广或提升 OSS，以低费用或外购替代者贡献到相应的项目。
  - 为 OSS 产品提供有偿服务，可能包括以合同形式为用户提高或修订产品的功能。
  - 调整商业产品成为 OSS。这可以提高产品的接受度，为公司创造新的商业机会。
  - 在双重许可下提供一个产品。在这样的案例中，一个产品在商业许可和作为 OSS 的情况下应用。OSS 的许可通常受限于某些方面（例如不允许集成到商业产品中），而商业许可以克服这些障碍。通过这样的方式，公司可以开拓团队的利益（例如外部的贡献者、大的用户群），并且仍然可以提供额外的有偿服务。

总之，许多 OSS 项目至少在一定程度上是由商业考虑驱动的。这种趋势在随后几年会延续下去，并支撑着 OSS 利润的增长。

法律层面

OSS 是成本（许可）免费的，但不是任何责任免除的。因此，当采用 OSS 开发时需考虑到法律层面。当讨论法律层面时，我们需要严格地区分合同方面（例如质保期）与知识产权（专利）方面的内容。

让我们首先看一下典型软件用户的看法：在目标 OSS 可以获取、安装和运行的情况下（通常基于 Linux、MySQL 或 Apache），并且不改变软件，法律层面和外购的商业软件就没有多少区别了。即使软件的源代码因为功能的需要做了改变，但只要没有分发给其它用户，通常没有法律层面的影响。但是，用户应该意识到一旦使用 OSS，就可能产生合同关系，包含了合同和版权两个方面。作为商业软件，购买者应确保供应商获得了第三方的免费使用权。

软件供应商或集成商的看法是：相对于正常的 OSS 用户，更复杂的情况是：当 OSS 用于开发新的将来发布的软件基础时，需要考虑 OSS 的许可条件和另外的特定国家的法律条款。取决于 OSS 许可模式，分销商承担一些 OSS 的特殊义务。例如，BSD 类型的许可（例如 Apache 服务器）允许 OSS 软件的修改和商业销售，但不能披露源代码。从另一方面讲，GNU 通用共用软件要求（修改的）源代码公开，嵌入到 OSS

的软件源代码也需要发布。如果仍然有疑问，那么就需要考虑专业的建议了。

开发过程

典型的 OSS 开发过程包括一组松散合作的开发者，他们同时在新的功能或改进方面工作。在许多案例中，需要使用版本化的系统（如 CVS 或其子版本）管理产品的源代码。通常，仅仅有少数可信任的开发者（叫做维护者）可以访问源代码库进行编辑，他们能够修改主要的开发程序。然而，其他开发者可以得到源代码的拷贝（例如，通过公开的、只读的路径进入代码库）并基于这些拷贝开发他们自己的应用。他们将包含修改的文件（文件补丁）递交给维护者，维护者审核，如果程序是正确的，就可以集成到主代码分支中。

分支代码可以导致独立的开发流程。在随后的过程中集成到主开发程序或分支形成一个新的项目（分叉）。

成熟的 OSS 项目通常同时存在许多流程来支持不同的活动，例如需求管理、放行管理、发布报告和跟踪、软件分发、软件测试，以及前面提到的版本和组态管理。这些过程通常由软件工具和团队队活动结合起来强化，团队的开放度和透明度是其中的关键因素。

支持和维护

一个 OSS 产品的积极团队也可以增加产品获得免费支持的机会。在许多案例中，OSS 产品的用户在开放的邮件清单或网络论坛中提出产品的相关问题；也有许多项目提供了一个问题跟踪系统。在这个系统中，每个人都可以报告问题或提出改进建议。无论如何，都必须考虑到不能保证这些免费的支持渠道一直工作。送到邮件清单的问题可能得不到回答，或者报告的问题也可能相当长的时间没有得到解决。原因包括技术工程师没有回答，团队中没有人知道答案，或者甚至认为需求或问题是不相关的。即使在一个大的项目中，一个积极的团队也会发生这样的事情。

对于需要支持保证的组织，仍然有一些办法，例如跟公司或者一些自由开发者签署合同，或者雇佣有经验的开发者或系统管理员在线提供支持。

客户-供应商关系

对比商业软件已知良好的客户-供应商关系，OSS 提供了几种供应商交互的方法，典型的方法如下：

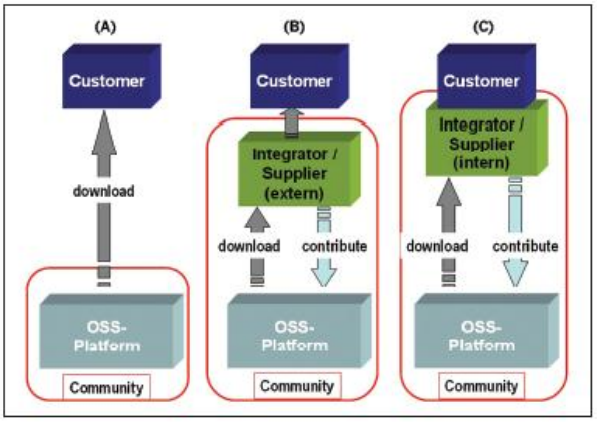

图 1-典型的客户-供应商情形

- 情形 a：客户下载、安装和使用 OSS 应用，应用并没有被用户或 IT 部门改变。在这种情形中，客户不是团队中的一部分。
- 情形 b：应用由系统集成商或其他供应商支持和维护（也可能是定制化）。根据许可模型，集成商或供应商可能是团队中的一部分并编写代码。
- 情形 c：像情形 b 一样，但是客户的 IT 部门作为集成方支持和维护应用，客户或客户的 IT 也可以修改代码。在这里，根据许可模型，客户甚至可以成为团队中的一部分。

这是三种典型的情形，在现实中，也可能存在不同的变化和组合。

从上述情形中的选择一种并制度化是非常重要的，由于他们牵扯到 OSS 客户的角色。例如，如果 OSS 软件减少缺陷和增加功能是非常重要的，可以建议针对该项目聘用外部的供应商（情形 b）或安排内部的开发者（情形 c）以保证问题的快速解决。进一步讲，如果 OSS 产品的修改给到第三方（例如，作为嵌合系统的一部分），根据 OSS 许可情况，甚至也可能需要调整团队。尽管第一眼看过去，投资潜在有益于其它公司的事情有些奇怪，但是接受来自于 OSS 团队的支持和产品改进还是值得这种付出的。

发布渠道

对于发布来说，OSS 的开发者的以下几种渠道都是可以的。

最通常的情况包括：

- 直接网络下载。在某些情况中，OSS 项目提供和管理他们自己的网络服务器（通常有来自公司赞助商的财务支持）。尽管许多项目依赖于网站提供通用的 OSS 项目服务，例如 SourceForget 或者 Launchpad，但通过这种普通渠道发布的软件，通常也仅仅在源代码的论坛上可以获取。
- 操作系统发布。对操作系统的内核（例如 Linux 或 BSD）和软件的公用及应用部分进行联合发布，可以提供一个可使用的、集成的系统。由于发布的开发者（OSS 项目就是他们自己发布操作系统）通常使不同的部件在一起工作，发布者通常最熟悉安装开放代码的产品。并且，在大多数的情况中，发布者为通用的硬件结构提供预编译的、可运行的代码。例如 Linux 系统的发布者是 Fedora、SUSE 企业和 Ubuntu。
- 网站软件下载。专业提供软件（例如 Tucows）下载的网站（通常是商业的），可以提供 OSS 产品不断增长的选择。
- 软件收集。软件收集，例如由计算机相关杂志发布的东西，通常也包含 OSS。

这些发布渠道相关的主要风险是无意下载到可能被第三方恶意修改的 OSS 产品。可以通过使用“官方”的服务器或平台下载，使这些风险最小化，因为许多情况下这些下载由校验或数字签名来确认其真实性。

基于 GAMP 的 OSS 系统生命周期

不管是商业软件还是 OSS，在 GxP 环境中都会运用同样的监管规则。

达到或维持 GxP 法规系统的合规性包含了 OSS 的部件与商业软件一样，能普遍地按照 GAMP 指南来完成 。对于 OSS 生命周期方法的不同和改进之处将在本章说明。在 GAMP5 中描述的总项目生命周期可以作为一个通用的框架。

类别	软件类型	示例	生命周期方法
1	基础软件	主要商业化 Linux 发布的核心系统和公用部分，如嵌入系统（例如固件、网络软件、Linux、BSD 操作系统、Apache HTTP 服务、MySQL）和建立的层级软件（例如 OpenOffice、Firefox、DIA）	标准的 GAMP 方法、只要合法的实体存在以维护软件，例如在 GAMP 指南中，为 SDLC 提供的服务能够考虑为供应商。
2	不适用	不适用（以前的固件，在 GAMP5 中已删除）	不适用
3	非组态软件	RANDI2	标准的 GAMP 方法。只要合法的实体存在以维护软件，例如，在 GAMP 指南中，为 SDLC 提供的服务能够考虑为供应商。取决于风险，供应商评估也许是不同的，参见章

			节 3.3。
4	组态软件	内容管理系统（CMS）、企业管理软件（CRM/ERP）	标准的 GAMP 方法。只要合法的实体存在以维护软件，例如，在 GAMP 指南中，为 SDLC 提供的服务能够考虑为供应商。按照 OSS 列出的标准进行供应商审核，范围和深度应取决于产品的使用目的。
5	订制软件	作者目前不知道来自于 OSS 订制的直接支持制药流程的软件。自定义的部分加入到一个项目中应考虑为订制代码。如果这样的软件经初始的测试和确认后作为 OSS 产品的一部分而发布，分类的结果应降低。	标准的 GAMP 方法用于 OSS 应用的验证

表 A-生命周期方法

表 A 列出了 GAMP5 的分类和 OSS 应用示例。在最右面的一列，给出了不同生命周期活动的一些建议。

概念阶段

概念阶段的目的是准备项目使之具备合适的资源。

项目/验证计划

项目计划定义了待开发的工作产品；生命周期使用的模型和方法；和项目管理相关的客户需求；需完成的任务；任务的所有权；项目资源；计划、里程碑和目标日期；预估值和质量标准。此外，计划识别的关键附属物；需求的工作产品；项目风险和风险转移计划；未完成任务的应急行动。

对于任何的验证项目，验证计划至少应该包括重要的背景信息、项目目标、责任人、需遵循的 SOP 描述、相关过程的标准和原则；以及用于结论的预先定义的接受准则。

需求/说明书

根据软件开发使用的生命周期，说明书可以用不同的方法实施。充分定义的需求说明书应该是有效的。说明书应该用输入和结果描述由软件支持的过程。当然，详细的程度也取决于 GAMP 类别及涉及的风险、复杂性和新颖性。技术说明书，像功能说明书或设计说明书一样，应该和选择的生命周期模型一致。

项目阶段

承包商/供应商评估

由于 OSS 的特殊性和不同的客户-供应商关系，承包商和供应商评估可能是非常有挑战性 的任务，相比于商业软件有很大的不同。

对于低风险的应用，供应商的评估是不需要的。对于中等风险或高风险的应用，推荐采用情形 b 和情形 c，因为一个服务组织-无论是内部或外部的-将减少不确定性以及团队的支持和维护风险。

对于情形 b，供应商评估和商业软件是一样的。对于情形 c，使用内部质量标准并且内部供应商需要按照情形 a 评估团队。评估的严格程度应与风险优先级一致。

如果一个团队被选择作为直接的资源（情形 a），合适的质量标准用来衡量团队必须具有的稳定性和质量。GAMP5 中提供了通用的 OSS 评估检查清单，并附上了材料（检查清单和问卷示例）。

OSS 团队的持续性能够使用下列的通用标准评估，当进行供应商评估时应考虑选择 [4]。标准的选择是取决于具体情况并应记录下来。

- 团队的活动
  - 下载的数量
  - 开发者的数量
  - 发布的数量
  - 邮件列表的活动和论坛
- 个人的档案资料
  - 关键开发者/维护者的经验（对于 OSS 开发者的一个重要动因是获得团队的认可）
- 沟通
  - 邮件列表
  - 新团队
- 团队内的组织结构[1]，例如：
  - 项目管理
  - 核心团队定义
  - 子项目的维护
- 组态管理
  - 适用审核的过程定义和进入主分支的集成
  - 明确从稳定放行产品中分类的流程
  - 版本控制
  - 系统和放行文档

额外的标准可以找到，例如 FlossQuality[14]。

开发标准

开发标准对于 OSS 软件是主要的挑战，但是实际上许多团队有现成的标准。根据许可模型，下列标准对于重新使用代码和修改代码是必需的，这应该是供应商评估的准则。

可追溯性

在 URS、功能说明、开发说明和测试之间的关系应该是可追溯的。例如如果你雇用团队的一部分来扩展或建立一个应用，你或团队被聘用的合同者必须提供按照 GAMP5 描述的方式提供追溯性。

风险评估

风险评估可以在不同的级别上执行。高风险级别的总体风险评估可以把系统的复杂性和风险作为生命周期活动策略的基本输入。

对于复杂的系统，在功能层次上的包括组态和代码的详细风险评估，对于达到目标或减少测试的付出是有用的。作为风险评估方法，EN60182 （描述了故障模式和结果分析（FMEA）和故障模式、结果和重要性分析（FMECA））以及 GAMP5 均可应用，后者应用于功能风险评估方法。但是故障树分析和其它方法（参考 ICH Q9）也是可以接受的。无论如何，记住总体的思路不是测量风险，而是识别和管理风险。

同样地，需要记住商业软件和 OSS 的风险评估方法是一样的 ——因此失败的后果也一样——但是随后的风险管理过程中风险转移方法有一些不同。例如 OSS 额外的代码审核是一个转移方法，但通常不能应用于商业软件，而采用对系统高度关键的功能部分增加测试的方法。

实施

实施阶段的目的是建立系统；因此，这个阶段仅应用于 4 类和 5 类的软件。在这里，商业软件和 OSS 的不同是明显的，应考虑采用不同的许可模式（参考 0 部分）：

- 如果你直接使用团队提供的软件（情形 a），这最不像是 5 类软件。从负责任和可靠性考虑，可能需要进行额外的风险管理。
- 如果你通过供应商购买 OSS（情形 b），也许包括团队提供的源代码或组态修改，你应该运用和商业软件同样的过程和测试。
- 如果你自己修改或组态软件（情形 c），当向其他人发布软件（公共版权）时，你也许是（变成）团队的一部分。如果

你只为自己的目的使用软件，版权问题就不存在了。对于第一种情形，你必须遵守团队的规则，在任何情况下，你对自己的工作，应改明确自己的内部流程和测试。这也许包括提高员工 OSS 的版权意识（见章节 2.2）和维持合适的许可文档。建议你检查软件的变更、扩展、或重新发布是否在计划内的。在 GPL 许可类型的软件部分，这是非常重要的。

测试的深度和严格程度应该与识别的风险相适应。

安装

安装过程比安装确认（IQ）包含更多的活动：你必须对安装过程有描述，包括预定条件和硬件就位等等。IQ 也能遵循这个过程和文件，只要它也像其它软件一样正确地执行。

接受测试

无论你是否遵循 V 模型或其它任何类型的开发模型，除了第 1 类型的软件，接受测试必须包括所有类型的软件，因此也适用于 OSS。对于 OSS，应用和商业软件同样的方法执行和记录接受测试。

培训

像所有的系统一样，用户和系统管理员必须培训以适合使用系统。推荐使用内部或外部的培训资源执行需要的培训。

验证报告

在项目结束时，系统需交接给操作人员。双方必须就系统应有的状态和系统的接受达成一致。合适的记录项目完成的事务形成验证报告。验证报告明确项目结束时从质量审核方面到系统放行使用的情况。放行的前提条件是成功地完成了验证计划中规定的事务。OSS 的验证报告与其它系统没有什么不同。

运行阶段

系统运行阶段的目的是给用户提供一个可运行的系统。

服务平台和事件

对于所有的应用，包括 OSS，应该定义一个有责任的人员或组织作为联系的单点。和商业软件相似，这个人员/组织应该管理事件。由于团队不一定有，甚至将来可能会解散，一个基于团队的服务平台就会造成额外的风险。因此，需要绑定内部资源或有合同关系的供应商。

对于 GAMP 高级类型和高风险的系统，有合法合同关系的供应商（SLA）应该是明智的选择。从 GMP 观点来看，低风险的系统可以在没有明确服务的情况下使用。对于两者之间的系统，服务可以是没有法定合同关系的团队提供，但需要执行一个评估。

像 SLAs 一样的合同应该和给出的标准保持一致，例如 COBIT，这与商业软件没有什么不同。

偏差/问题管理

偏差和问题管理的处理应该有清楚的定义。尽管团队可以提供这些问题的解决方案，但是事件的跟踪和分类的管理只能由供应商完成。因此，推荐成立内部资源或建立有合同关系的供应商。

当在一个团队中工作，一个错误程序跟踪系统以及功能和需求的支持记录也是必须可获取的。这些沟通机制必须和你的内部或外部服务供应商整合在一起。

变更和组态管理

变更和组态管理发生在两方面：

- 客户/用户：像其它的应用，合规的公司必须实施 OSS 应用的变更管理系统。
- 供应商/团队：供应商或团队需要提供足够的详细数据允许客户/用户方面的变更管理，以及允许软件的变更可以跟踪。取决于系统的复杂度，可以使用不同的方法，例如补丁放行事项或源代码评注均是足够的。建立变更管理流程来管理和控制变更。

无论是内部还是外部的服务组织都必须监督团队的活动。错误程序的修正和放行必须得到评估和适当实施。因此对比与商业软件，总的方法是不同的，但是 OSS 更新可以有更多的细节，并用风险评估支持。

维护验证的状态

一般而言，有两类的评估：

- 改变系统后：作为变更控制流程的一部分，每一个变更的影响和风险都需要确认、评估和管理。
- 周期审核以确保当前的系统能力。

和其它的系统一样，周期审核的频度和范围应该用基于风险的方法和并由历史数据的回顾来决定。

退休阶段

再一次，系统的退休对于 OSS 没有什么不同。最重要的需求是管理和提供保留周期内的数据。对于 OSS，数据格式和算法可以用来支持数据分析和转移。

总结和将来的趋势

尽管 OSS 看作和传统商业软件完全不同的软件，但对于监管行业的重要性证明差别很小。不同之处在于对许可模型和开发过程的影响，依次是供应商选择和服务过程的影响。无论如何，生命周期活动是基本一样的，GAMP5 中基于风险的方法也仍然可以应用的。

相应地，我们没有原因在监管的行业中排斥使用 OSS。OSS 应用越来越多地出现，使其可以替代商业软件。例如，开放文档格式（ISO/IEC26300）提供了独立的应用格式，并且为在不同的系统中存档和交换文件提供了一个完美的解决方案。到现在，仅仅 OSS 应用可以支持这种格式。

其它监管的行业像航空公司是领先的，他们已经在重要的领域中使用了 OSS。

参考文献

1. Dietzs, S., Modell und Optimierungsansatz fur Open Source-Software entwicklungs prozesse, 2003
2. Bell, B.S., et al., Times “R” are A Changing, FDA Perspectives on Use of “Open Source,” 2006, http://www.fda.gov/cder/offices/biostatistics/Bell.pdf
3. Pharmaceutical cGMPs for the 21st Century – A risk-Based Approach, Final Report – Fall 2004, http://www.fda.gov/cder/gmp/gmp2004/GMP_finalreport2004.htm
4. Ripamonti, L.A., DeCindio, F., and Benassi, M., “Online Communities Sustainability: Some Economic Issues,” The journal of Community Informatics, Vol. 1, No 2, 2005, p. 6378, http://ci-journal.net/index.php/ciej/article/view/221/180.
5. WHO Expert Committee on Specifications for Pharmaceutical Preparations, 2006, p. 143
6. Volume 4 – EU Guideline to Good Manufacturing Practice – Medicinal

Products for Human and Veterinary User, 2008, p. 111ff
7. 21 CFR, Electronic Records, Electronic Signatures, 2008
8. 21 CFR 211 Current Good Manufacturing Practices for Finished Pharmactuticals, 2008
9. Guidance for Industry – Part 11, Electronic Records; Electronic Signatures – Scope and Application, 2003.
10. ICH Q7, Good Manufacturing Practice Guide for Active Pharmaceutical Ingredients, 2000, http://www.ich.org
11. PIC/S Guidance Good Practice for Computerized Systems in Regulated "GXP" Environments, 2007, http://www.picscheme,ord
12. EN 60812:2006-11, Analysis Techniques for System Reliability – Procedure for Fialure Mode and Effects Analysis (FMEA).
13. Flossquality, http://www.flossquality.eu/

ISPE GAMP 5: A Risk-Based Approach to Compliant GxP Computerized Systems, International Society for Pharmaceutical Engineering (ISPE), Fifth Edition, February 2008, Appendices D5 and M5, http://www.ispe.org.