

《企业级开源软件 评估评价体系探讨》

龚 仪

现代软件工业从创立之初就围绕着开源分享与协作



2008年 软件工业的大多数角落都出现了开源软件的身影
2011年 低成本已不再是市场接受开源软件的第一因素
2013年 如果软件正在吞噬世界，开源软件正在吞噬软件世界

开源软件生态体系

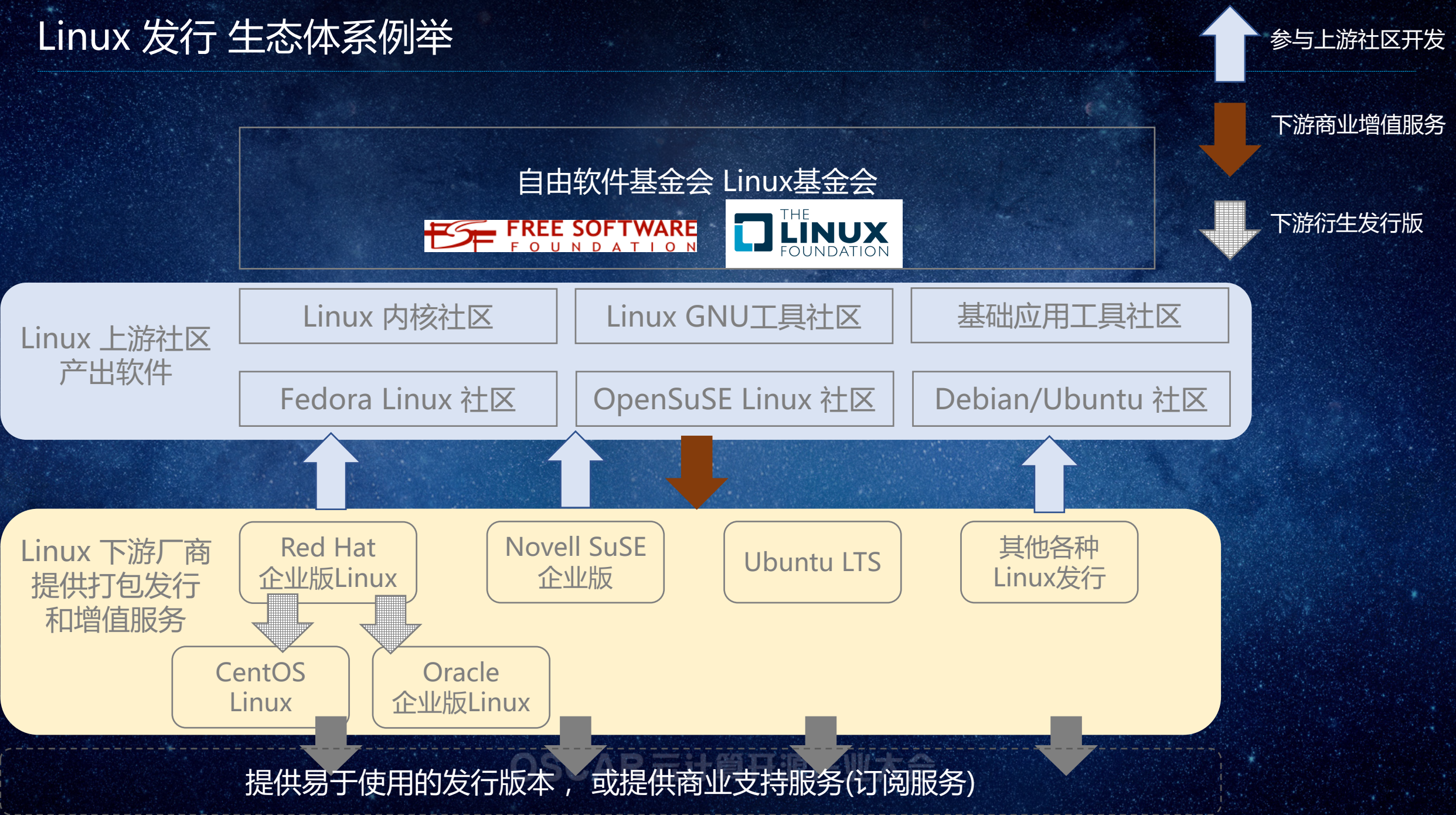
- 权威合法组织
- 政府接口中心
- 财/才 资源中心
- 法律，规范中心
- 厂商接口中心

- 委员会形式管理
- 代码审核权控制
- 代码递交权控制
- 成熟的全球化联合开发协作模式
- 核心开发成员由对口基金会聘用

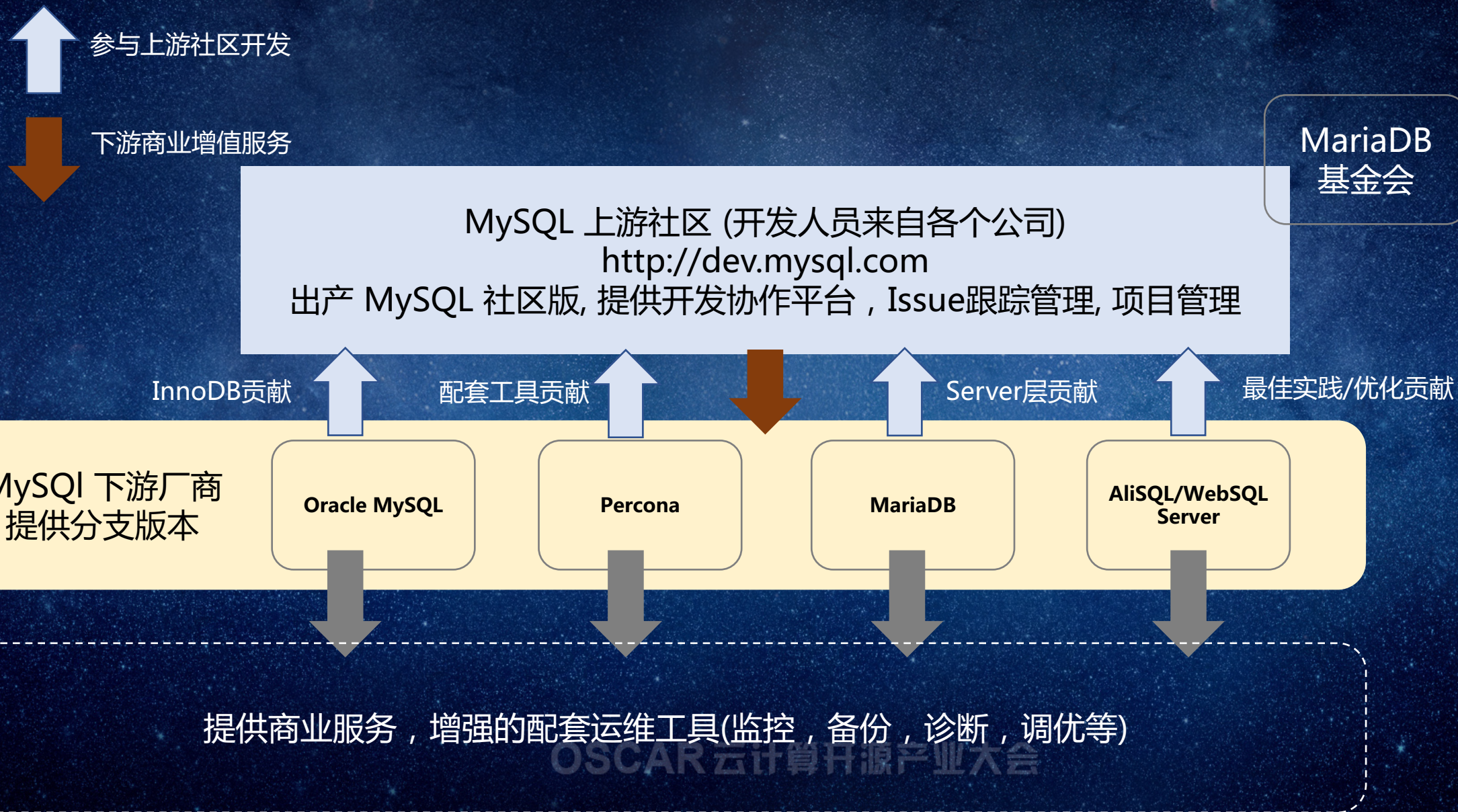
- 开源软件产品化
- 以订阅模式交付商业开源产品
- 与主流社区保持紧密联系和互动



Linux 发行生态体系列举



MySQL生态体系列举



开源软件的特点

变化快，产品分支多

开源软件通过源代码开放给全球开发者和企业。

- 产品迭代速度快，版本变化很快
- 同一类产品，有不同的分支版本
- 不同类产品具有不同的许可证

开源软件的研发特点:

- 重视核心算法，核心功能的研发
- 软件的配套管理功能普遍较弱

重核心功能，轻配套管理

社区及基金会主导软件项目管理

不同于闭源软件的企业内部研发管理:

- 开源项目一般由上游社区负责研发
- 项目和社区的治理通常由基金会提供资金，基础设施，人员和法律方面的支持和治理

开源软件的技术支持由上游社区提供:

- 无SLA保障的互联网支持模式
- 支持对象以开发者为主，使用者为辅。

社区技术支持对象以开发者为主

开源软件

企业应用开源软件问题和风险

1 缺乏科学的准入和选型体系	2 缺乏开发测试及运维标准	3 缺乏最佳实践指引和方法论	4 缺乏人员培养和知识管理
主要风险和问题 开源软件组件和架构选择以满足业务功能为主，没有紧密结合运维管理要求及未来可持续运维能力建设的要求 准入体系缺失将导致配套管理成本极速增加，开源软件的成本优势被削弱。运维安全风险显著增加。	主要风险和问题 没有适配开源软件特点的在开发测试的应用标准和生产环境的运维标准。 标准化缺失将产生规模化应用后的管理风险，同时阻碍了现代数据中心自动化和智能化管理能力的提升。	主要风险和问题 开源软件带来的灵活架构能力是双刃剑，是需要最佳实践指引和配套方法论保驾护航的。 缺乏对标业务和管理需求的开源架构及业界最佳实践的参考，开源软件的“灵活架构”将成为一个管理灾难。	主要风险和问题 技术团队除了要掌握开源软件技术本身之外，更重要的熟悉和掌握开源社区的信息获取，筛选，判断，互动，反馈和整理这6项基本能力。 只掌握开源软件技术本身而缺乏与开源社区的连接，将导致企业成为开源软件应用的“孤岛”。

主流开源软件评估模型的研究

模型缩写	模型名称	发布时间	来源	组织	含评价方法
C-OSMM	Open Source Maturity Model	2003	Duijnhouwer & Widdows, 2003	Cap Gemini	是
O-BRR	Open Business Readiness Rating	2005	Wasserman, Chan, & Pal, 2005	Open-BRR	是
N-OSMM	Open Source Maturity Model	2005	Golden, Making open source ready for the enterprise: The open source maturity model, 2005	Navica Software	是
Q-SOSS	Methodology of Qualification and Selection of Open Source	2006	Atos-Origin, 2006	Atos-Origin	是
Q-QMM	Open Source Maturity Model	2009	Wittmann & Nambakam, 2009	QualiPSo	否

注1: Cap Gemini – 凯捷安永 全球著名的咨询公司
注2: Open-BRR – 由卡内基梅隆大学, 著名的软件测试公司 SpikeSource, 由最大的开源出版机构O’ Reilly和Intel 共同成立
注3: Atos-Origin 全球知名的大型IT技术和咨询公司

主流开源软件评估模型的研究 – 分析

以CMMI(软件能力成熟度模型集成)中的软件评估作为参考，深入研究了当前主流的开源软件评估模型，我们得到以下重要的结论：

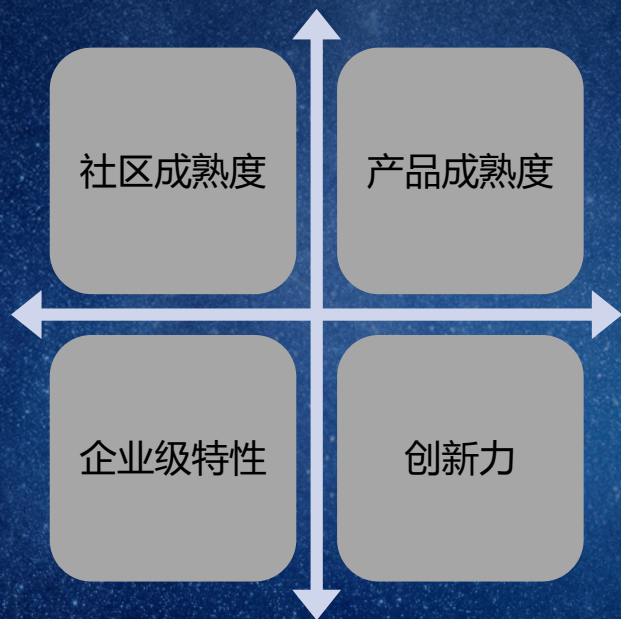
- 1.原有的开源软件评估模型都出现的比较早，而当前开源软件的发展随着虚拟化，云计算，大数据，分布式等平台级技术的迅猛发展，导致这些模型的评估体系，评分体系和权重设计都无法完全精准和高效评估要求。
- 2.原有的开源软件评估模型，大多停留在学术研究层面，无法适应工程化和商业化对于评估本身的高效，低成本和实效性的要求。
- 3.开源社区的软件设计和开发协作已经从较早的个人开发者兴趣驱动全面转向商业化驱动的发展形态，原有评估模型缺乏对当前开源软件驱动力本质的判断和评估。
- 4.当前开源社区和开源项目的发展已经从传统较为集中式的社区开发邮件列表+Bug跟踪转向了以Github 为代表的SNS 开发协作模式，原有的评估方法的数据采集和输入项无法精确跟踪现有开源开发模式的转变。

主流开源软件评估模型的研究 – 结论

我们需要在原有评估模型的基础上，取长补短，设计出符合工程化和商业化的精准和高效的开源软件评价体系，需要实现如下方面：

- 1.能够高效，精准，低成本的完成对现代开源软件的评估。
- 2.符合现代开源软件商业驱动，规模化，SNS(社交化)的发展特点。
- 3.评估体系中的三要素(评价指标，打分规则，权重规则)需要满足企业场景而非研究用途。

开源软件评价体系的形成思路和设计成果



维度解读:

- 当前主流开源软件虽然由商业驱动但仍旧缺乏商业厂商支持，对于用户来说，社区的成熟度至关重要，社区成熟度的高低，直接决定了使用开源软件的风险和成本的高低。
- 产品成熟度决定了开源软件是否进入工程化和商业环境安全可靠运行的基本准入条件。
- 企业级特性要求开源软件在可靠性，可用性，可管理性，可扩展性上符合和满足企业级应用的要求。
- 创新力关系到采用开源软件，尤其是基础架构和平台级开源软件的过程中，能否保持技术演进速度和优势的持续能力，也显著的影响到应用开源软件的风险和成本管理。

企业级开源软件评估体系 E- OSMM (Enterprise Open Source Maturity Model)

评价项	评价指标	评价方法	覆盖维度
I. 项目基本面评估	项目年龄	项目正式发布至今时间	社区成熟度 创新力
	项目的卖点	项目发起者宣称要解决的核心问题及技术亮点	
	许可证类型	GPL/BSD/Apache等	
	发布模式	只有社区版还是同时存在商业版	
	版本控制	ReleaseNotes/CHANGELOG 发布质量	
	支持平台	支持的OS平台, 软件包格式种类	
	开发协作模式	github/launchpad/google code/sourceforge/其他	
	发布频率	大版本之间和小版本之间的发布间隔和频率 (次/年)	
	发布流程	开发代码树的递交和合并规则, 代码树的管理规则	
	产品质量	Bug/Issue库的统计, ReleaseNotes/CHANGELOG的故障分级统计	
II. 产品基本面评估	产品标准化	是否满足和遵循一种或多种行业标准化	社区成熟度 产品成熟度
	产品模块化	产品自身是否存在模块化设计或以模块化方式集成	
	开发人员规模	代码递交统计数(公司内和社区)	
	社区活跃度	社区核心协作平台:如邮件列表/github上的活跃度	
	社区影响力	社区活动的途径方式, 社区市场活动的频率, 次数和规模	
	社区控制力	社区决策层的主要构成和决策规则	
	商业市场影响力	与商业团体等第三方商业生态圈的合作程度	
	服务形式	社区支持还是有商业支持, 分别以何种方式提供	
	服务内容	服务内容对于SLA的覆盖程度	
	服务响应能力	社区支持的响应能力, 商业支持的响应能力	
III. 服务与支持评估	服务本地化	本地化社区, 本地化商业支持团队, 本地化沟通及服务	社区成熟度 企业级特性
	服务影响力	社区支持服务看社区支持的活跃度, 商业支持看订阅服务购买情况	
	服务的成本	社区服务的成本构成, 商业服务的成本构成	
	可靠性	开源软件的可靠性设计和实现(通过基础评估测试)	
	可扩展性	开源软件的可扩展性设计和实现(通过基础评估测试)	
	性能	开源软件性能设计和实现(通过基础评估测试)	
	可用性	开源软件的可用性架构设计和实现(通过基础评估测试)	
	易用性	CLI(命令行交互界面), UI(图形化交互界面) 的完备性	
	平台独立性	对平台和运行环境的依赖性	
	安全性	已包含和提供的安全相关的能力	
IV. 产品核心技术评估	可管理性	产品配套工具是否完善, 如备份, 恢复, 监控等	产品成熟度 企业级特性
	文档完备性	产品规格文档, 管理文档, 配套组件文档, 在线教程, 其他文档资源	
	故障库完备性	是否提供故障库, 故障库的信息含量和完备程度	
	培训	社区培训或商业培训, 是否有认证制度, 培训内容覆盖面	
	供应商独立性	是否存在专业商业供应商或第三方独立供应商	

输入您的主题

OSCAR 云计算开源产业大会

输入您的主题

OSCAR 云计算开源产业大会

THANKS