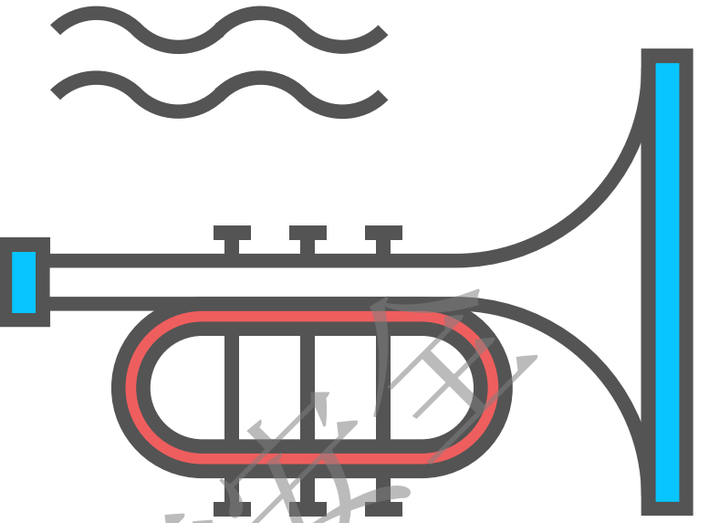
# 《开源软件成熟度评估白皮书》发布及解读

国家工业信息安全发展研究中心软件所

# 一 背景

**开源软件**即指源代码开放的软件，遵循开放软件源码、软件可自由发行、允许衍生、保护源代码的完整性、对用户和使用领域无差别对待以及技术中立等原则。



- 开源是软件产业创新的重要模式，引领软件技术演进、产品升级、产业发展
- 开源已成为全球软件产业生态的重要组成
- **如何选择开源软件？**
- **怎样评估开源软件？**

# 二 开源软件成熟度评估的意义

开源软件成熟度评估旨在对开源软件的属性进行成熟程度评估，有助于提高开源软件的质量，促进开源软件的推广应用。

01 可为选择符合自身需求的开源软件提供依据

开源软件成熟度评估可以对软件的功能性、安全性、可靠性等多个方面进行评估，评估结果可以帮助用户、开源软件集成商、服务提供商等选择符合自身要求的开源软件。

02 提供开源软件技术架构、代码质量以及非技术因素等方面的反馈参考

对软件源代码的质量、可信度、软件社区的开发管理等因素进行综合分析，其结果对于软件设计者、开发者以及用户都有很高的参考价值。

03 为开源软件的推广、应用提供有效的建议

开源软件成熟度评估可对开源软件的组织形式、开发模式、运营模式及社区参与者等多个层面进行评定分析，对于开源软件在推广与应用过程中存在的不足，提出合理的建议。

# 三 开源软件成熟度评估主流模型

## 国际上五种主流开源软件成熟度评估模型

1

**OSMM Capgemini**

时间：2003年
提出者：Capgemini公司

2

**OSMM Navica**

时间：2004年
提出者：Navicasoft公司

3

**QSOS**

时间：2004年
提出者：Atos Origin公司

4

**OpenBRR**

时间：2005年
提出者：Carnegie Mellon Silicon Valley SpikeSource，O’Reilly，英特尔公司

5

**OMM**

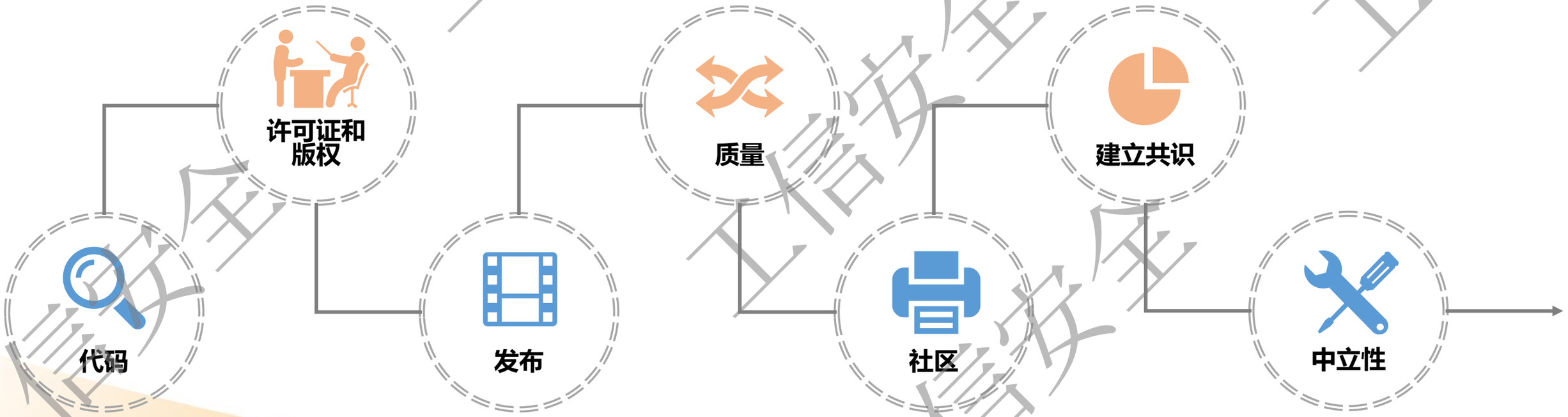
时间：2008年
提出者：欧盟QualiPSo项目

# 三 开源软件成熟度评估主流模型

| 评估模型 | OSMM Cappemini | OSMM Navica | QSOS | OpenBRR | OMM |
|---|---|---|---|---|---|
| 提出时间 | 2003年 | 2004年 | 2004年 | 2005年 | 2008年 |
| 创始人 | Capgemini公司 | Navicasoft公司 | Atos Origin公司 | Carnegie Mellon Silicon Valley SpikeSource，O’ Reilly，英特尔公司 | 欧盟QualiPSo项目 |
| 授权许可 | 非免费许可，但被授权运行发布 | 评估模型许可：Academic Free License | 评估方法与结果许可：遵循GNU Free Document License（GFDL） | 评估结果许可：遵循Creative Commons License（CCL） | Creative Commons Attribution-Share Alike 3.0 License |
| 模型用途 | 实际应用 | 实际应用 | 实际应用 | 科研 | 科研 |
| 分级模式 | 两轴，每轴分两级 | 分三级 | 三级或以上 | 分二级 | 分三级 |
| 是否有预定义规范 | 是 | 是 | 是 | 是 | 是 |
| 技术/功能规范 | 无 | 无 | 有 | 有 | 有 |
| 评分模式 | 灵活 | 灵活 | 灵活 | 严格 | 灵活 |
| 评分规范 | 1~5级 | 1~10级 | 0~2级 | 1~5级 | 1~4级 |
| 迭代过程 | 否 | 否 | 是 | 是 | 是 |
| 权重规范 | 是 | 是 | 是 | 是 | 是 |
| 结果比对 | 是 | 否 | 是 | 否 | 否 |

# 四 国外开源软件成熟度评估实践——Apache软件基金会

Apache项目成熟度模型给出的是一个成熟的开源项目应该具备的状态，而不是一组规则。该模型分别从代码、许可证和版权、软件发布、软件质量、社区、建立共识、项目独立性七个方面描述成熟的开源项目。



# 四 国外开源软件成熟度评估实践——CNCF基金会

## CNCF 项目成熟过程


### 沙箱阶段

- 采用CNCF行为准则
- 遵守CNCF知识产权策略（包括商标转让）
- 在网站/readme中显著列出沙箱状态

### 孵化阶段

- 满足沙箱阶段的要求
- 有至少三个独立终端用户在生产中成功使用了该项目，这些终端用户需要具备过硬的质量和能力
- 有足够的提交者
- 有大量的提交和合并贡献
- 有明确的版本计划
- 规范必须至少有一个公共参考实现

### 毕业阶段

- 满足孵化阶段标准
- 提交者来自至少两个组织
- 已获得核心基础设施计划最佳实践勋章
- 已完成了独立的第三方安全审计，所有关键漏洞必须在毕业前处理完
- 明确定义项目治理和提交者流程。将其放在GOVERNANCE.md文件中，并引用OWNERS.md文件，在该文件中给出当前提交者和名誉提交者
- 在主仓库要有项目采用者的公开列表，需列出规范实现的采用者列表
- 获得TOC的绝大多数投票进入毕业阶段。如果项目能够证明具备足够的成熟度，可以尝试从沙箱直接毕业。项目可以无限期保持孵化状态，但一般两年内毕业

# 四 国外开源软件成熟度评估实践——LF AI基金会

## 孵化阶段

- 1)项目使用OSI批准的开源许可证。
- 2)项目不依赖未经OSI批准的许可所专有或许可的组件。
- 3)通过Github拉取请求向TAC提交完整的项目贡献提案，并发送简短的电子邮件通知到info@lfai.foundation。
- 4)提供TAC或GB要求的其他合理信息。
- 5)可提供相当数量的正在进行的提交和合并操作信息。
- 6)拥有足够数量的提交者。
- 7)被TAC和GB认为可以为人工智能、机器学习或深度学习领域增加价值，并且属于LF AI的使命和范围。
- 8)同意将任何相关商标转让给Linux基金会或其分支机构LF Projects、LLC，并协助申请任何未注册的相关商标。

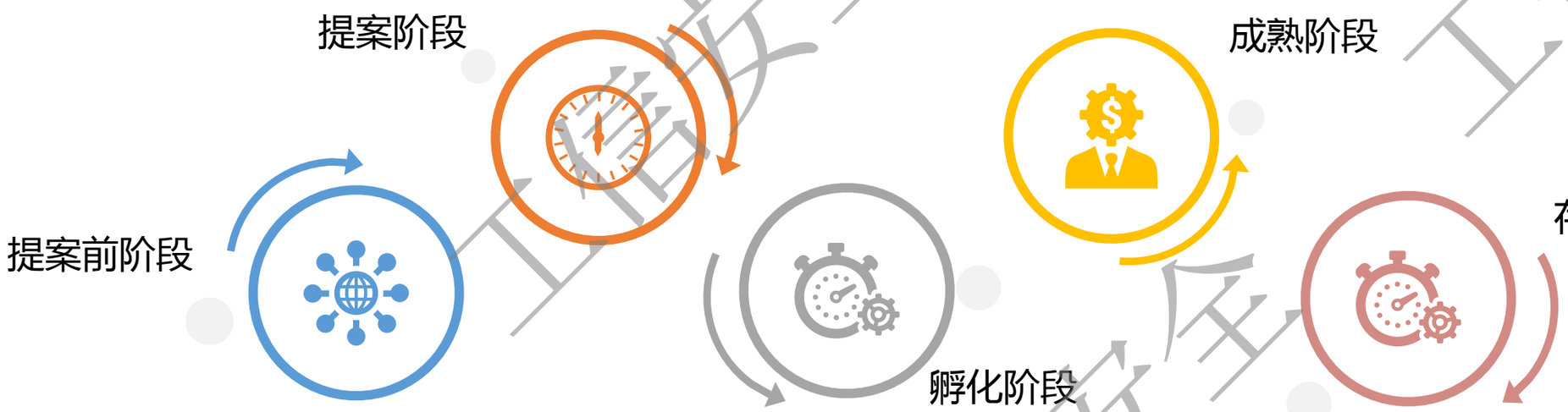
## 毕业阶段

- 除满足孵化阶段的要求外，项目还需具备满足如下条件：
- 1)提交者数量良好且至少来自两个不同的组织。
- 2)已获得并维护了核心基础架构计划最佳实践徽章。
- 3)可以展示大量正在进行的提交和合并操作。
- 4)在OWNERS.md和COMMITTERS.md文件中记录当前的项目所有者以及当前的和退休的提交者；项目章程（或其他授权的管理文件）的副本应包含或链接到GOVERNANCE.md或类似文件中。
- 5)获得TAC三分之二的赞成票和GB的赞成票。

## 退休阶段

- 退休项目是指维护人员认为已经达到或接近寿命的项目。
- 经TAC 2/3投票及项目所有权人批准，可授予项目荣休资格。在缺乏项目所有权的情况下，只需要TAC的2/3投票。

# 四 国外开源软件成熟度评估实践——Eclipse基金会




## 成熟阶段

- 1）该项目有一个足够高质量的可工作和可演示的代码库。
- 2）用一个活跃且足够多样化的社区适合于该项目的代码库，该社区应该拥有：采用者、开发人员和用户。
- 3）遵循Eclipse的原理和目的，完全在公开环境下运行。
- 4）该项目可以在较大的Eclipse社区中运行良好。

# 五 CIC开源软件成熟度评估模型（CIC-OSMM）——考虑要素

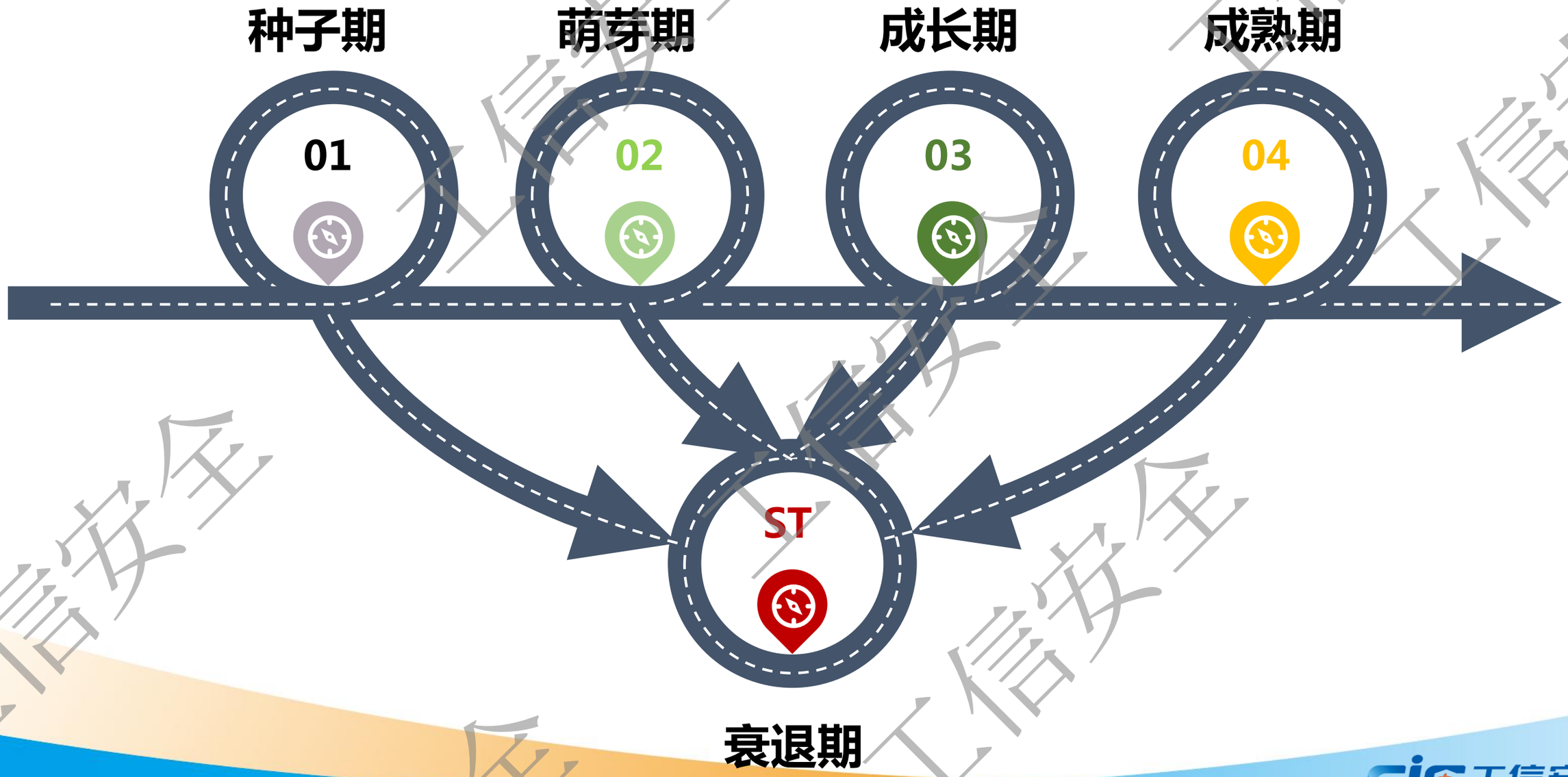
# 五 CIC开源软件成熟度评估模型（CIC-OSMM）——模型构建

## CIC-OSMM成熟度等级

CIC-OSMM分析归纳开源项目的成长规律，将其成长路线划分为五个阶段，形成CIC-OSMM的五个成熟度等级，分别为种子期、萌芽期、成长期、成熟期和衰退期。

- 1种子期
- 2萌芽期
- 3成长期
- 4成熟期
- ST衰退期

# 五 CIC开源软件成熟度评估模型（CIC-OSMM）——项目成长阶段



# 五 CIC开源软件成熟度评估模型（CIC-OSMM）——项目成长阶段

## 种子期 1

- 没有OSI许可或FSF许可下的源代码，或者没有允许第三方贡献的IP管理流程。

## 萌芽期 2

- 在公共版本控制系统中提供了可验证的源代码，源代码有可追溯的IP所有权和许可；
- 项目所有者具有一种机制，可以参与和理解第三方对项目的兴趣，第三方能够检查、理解并影响项目的未来；
- 项目有自己的问题跟踪器、信息网站和邮件列表等基础设施。

## 成长期 3

- 源码是根据OSI许可或FSF许可进行发布的；
- 代码具有比较合理的项目治理流程，能多次发布开源软件版本；
- 社区开始建立健全社区规范和知识产权管理策略，第三方愿意并且有能力对项目开发的主要方面负责；
- 项目的问题跟踪器、邮件列表处于活跃状态，能及时响应第三方请求；
- 社区进一步壮大，吸引更多的提交者和贡献者。

## 成熟期 4

- 项目社区具有完善的社区规范、项目治理流程和知识产权管理策略，并记录在册，社区能够严格遵照执行；
- 项目完全中立，不受任何公司或组织的控制和影响；
- 能够按照代码治理流程定期合规发布软件版本；
- 发布的软件发行版已经完成了独立的第三方安全审计；
- 提交者具有多样性。

## 衰退期 ST

- 项目所有者撤回项目，不再开源；
- 项目从未真正开始过;
- 缺少社区利益；
- 项目无法发展社区；
- 项目未能壮大社区规模；
- 项目无法建立活跃社区；
- 开发停滞；
- 社区活动（如邮件列表、讨论等）匮乏，社区不活跃；
- 项目虽然基本上完成了最初的计划，但后来没有找到可以让项目更持久的，并能继续发展社区的开发任务或目标；
- 项目影响力持续下降，能量逐渐消失。

# 五 CIC开源软件成熟度评估模型（CIC-OSMM）——模型特点

CIC-OSMM充分借鉴和吸收了国际主流开源软件成熟度评估模型和国外优秀开源软件成熟度评估实践理念。

CIC-OSMM充分考虑了开源项目的成长路径，并结合国内开源项目的发展状况，将衰退期作为项目评估的一个等级，有利于减少和筛选掉“KPI项目”，有利于推动国内开源软件的健康发展。

CIC-OSMM设置了合理的开源项目成熟度等级，并从源代码可用性、代码质量、知识产权策略、社区健康水平四个维度给出了各个等级应该具备的关键特征。

应用CIC-OSMM，不仅有利于帮助开源软件用户衡量社区健康水平、代码质量、知识产权风险等，指导用户做好项目选型，还有利于帮助开发者社区及早发现项目发展过程中忽视的工作，辅助社区健康发展。

# 六 未来展望

## 完善CIC-OSMM

继续完善CIC-OSMM，研究制定配套的评估指标体系和流程，并与相关机构合作，试点开展开源软件成熟度评估相关工作，验证模型有效性和科学性。

## 应用推广开源软件成熟度评估工作

鼓励和帮助开发者社区和用户应用CIC-OSMM开展开源软件成熟度评估工作，开源软件开发者社区和用户可以基于CIC-OSMM，采用项目自评与专家他评相结合的方式对开源软件项目进行综合评估。

## 加强开源生态治理，共建健康开源生态

将开源软件成熟度评估与开源软件许可证检测与代码安全测试工作相结合，从开源软件成熟度与开源软件合规、安全等方面对开源软件质量进行规范和引导，打造健康可持续发展的开源生态。

# 谢 谢！


国家工业信息安全发展研究中心

CHINA INDUSTRIAL CONTROL SYSTEMS CYBER EMERGENCY RESPONSE TEAM

地 址：北京市石景山区鲁谷路35号

通信地址：北京市750信箱