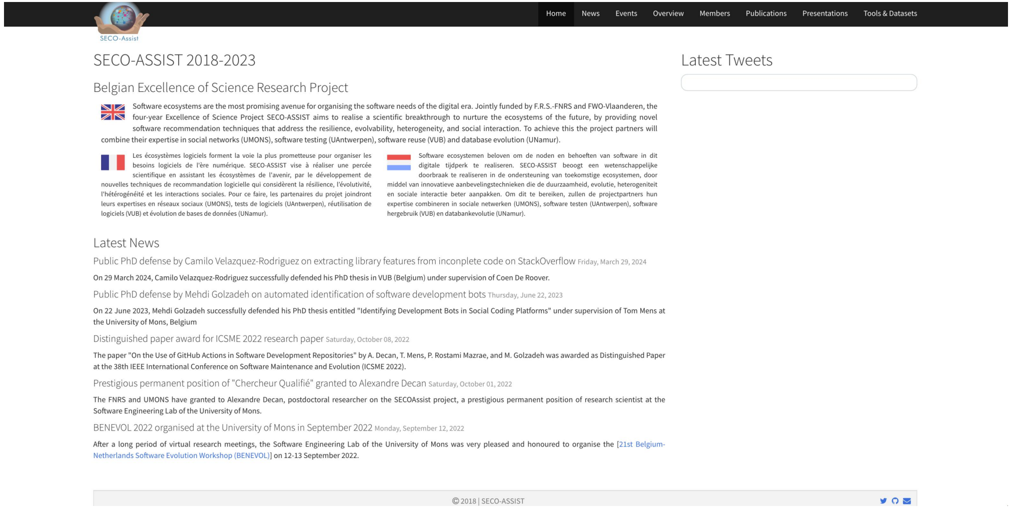
开源软件生态系统研究概论

报告人：毕枫林

参考文献

- Software Ecosystems Tooling and Analytics (Tom Mens Coen De Roover Anthony Cleve)
  - secoassist.github.io
  - 在实证分析中研究了与可重用软件库的软件生态系统相关的各种维护问题
- 其他论文
  - Software ecosystems – A systematic literature review（ Konstantinos Manikas ）
  - A Systematic Mapping Study on Software Ecosystems (Olavo Barbosa)



The Origins of Software Ecosystems

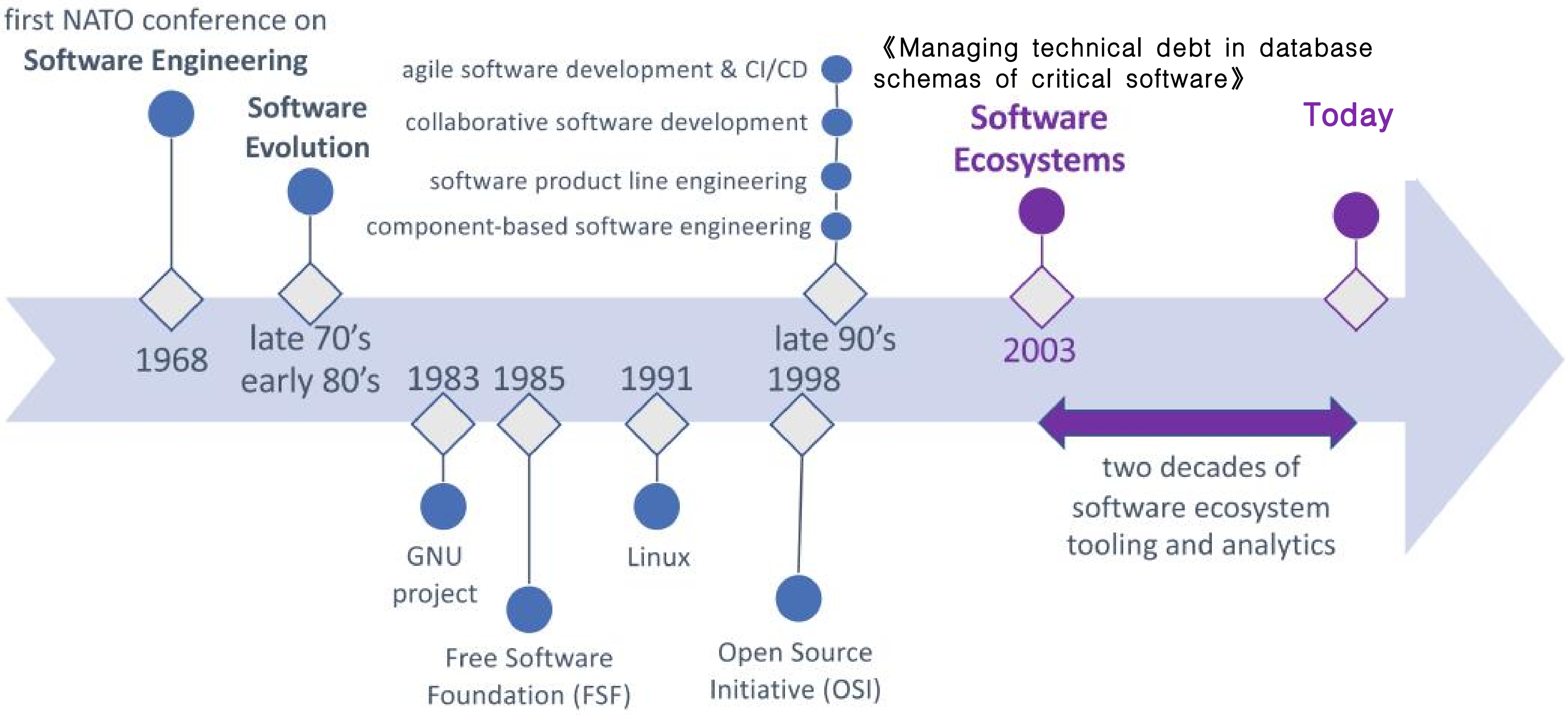


Fig. 1.1 Milestones that contributed to the domain of research (analytics) and development (tooling) of *software ecosystems*

Perspectives and Definitions of Software Ecosystems

1. 第一次提出

一组具有一定程度共生关系的软件产品

- Messerschmitt and Szyperski

2. 生态学视角（ecological perspective）

环境中的物理和生物组成部分被视为一个单元相互关联。

- Roy Clapham

3. 经济和商业角度（economic and business perspective,）

- 一组作为一个单位运作并与共享软件和服务市场互动的企业，以及它们之间的关系

  - Jansen

- 一个软件生态系统由软件平台、一组内部和外部开发人员以及服务于用户群的领域专家社区组成，这些用户群构成了相关解决方案元素，以满足他们的需求。

  - Bosch

- 一个由组织构成的网络社区，它们之间的关系基于对一种中心软件技术的共同兴趣。

  - Hanssen

Perspectives and Definitions of Software Ecosystems

4. 技术角度 (technical perspective)
定义软件生态系统为在同一环境中开发和共同演进的软件项目集合。

- Lungu

5. 社会视角（social perspective）
环境中的物理和生物组成部分被视为一个单元相互关联。

- Roy Clapham

6. 一个全包括的定义
一组参与者在共同技术平台上的互动，从而产生多个软件解决方案或服务。每个参与者都受一系列利益或商业模式的驱使，并通过共生关系连接到其他参与者和整个生态系统，而技术平台则以一种允许不同参与者参与和贡献的方式进行构建.

- Manikas

What are the main characteristics of a Software Ecosystem?

- 表2列出在研究中确定的软件生态系统的主要特征
- 关于开源软件的文献强调了各方之间合作建立成熟开源平台的重要性。
- 其他结果表明，软件生态系统与自然生态系统、软件产品线和商业生态系统有关联

- 表3列出了与软件生态系统相关的最常见术语和首字母缩写词。
- 自由、开源软件生态系统跟Software Ecosystems是相近的研究。
- Digital Ecosystem的方法论都是通用的

Table 2. Characteristics of a Software Ecosystem

Characteristics	Study ID
Inherit characteristics of natural ecosystems like mutualism, commensalism, amensalism, symbiosis and so on.	1,2,17
Is linked to business ecosystem perspective	5,12,22,25,29,35,42
Is linked to architectural concepts like interface stability, evolution management, security, reliability.	7,10
Is linked to Software Product Lines development model	8,32
Is linked to Open Source Development Model	9,14,16,17,28,27
Can be used to negotiate requirements for aligning needs with solutions, components, and portfolios	30
Collaborative Development in Government can be seen as a type of Digital ecosystem	34
Is related to innovation processes	1,3,4
Have their characteristics inspired by concepts like Software Supply Networks	5,15,41
Have impact on small and medium-sized Enterprises	36,40

Table 3. Terms related to software ecosystems field

Terms used explicitly	Study ID
Software Ecosystems	1,8,13,16,18,26,30,31,38,43,44
SECO (Software Ecosystems)	5,6,7,12,29
Mobile Learning Ecosystems / Mobile Ecosystem	6,11
Free Open Source Software Ecosystem, Open Ecosystem	9,13,31,34
Digital (Business) Ecosystem	33,34,36,39,42,43

Benefits and Challenges of Software Ecosystems Perspective

- 表4显示关于研究软件生态系统的好处。。
- 软件生态系统促进共同演化、创新和增加对新参与者吸引力的好处
- 软件生态系统使公司能够降低成本
- 支持架构决策制定
- 分享知识

- 表5呈现了软件生态系统视角中涉及的主要挑战和限制。
- 生态系统建模
- 架构挑战
- 许可证的异质性和软件演进

Table 4. Benefits of Software Ecosystems Perspective

Benefits	Study ID
Fosters the success of software, the co-evolution and innovation inside the organization, increase attractiveness for new players	1,2,4,7 9,10,23
Decreases costs involved in software development and distribution	3,10
Helps analyzing and understanding the software architecture in order to decide which platform to use	11,23,24
Supports the cooperation and knowledge sharing among multiple and independent entities.	20,33
Enables analysis of requirements communication among stakeholders	30
Comes as alternative to overcome the challenges during design and maintenance of distributed applications	16,36
Provides help to the tasks of business identification, product architecture design, risk identification	19
Provides information for the product line manager regarding software dependencies	32

Table 5. Challenges and Limitations of Software Ecosystems Perspective

Challenges and Limitations	Study ID
Establishing relationships between ecosystem actors and proposing an adequate representation of people and their knowledge in the ecosystem modeling.	2, 15
Several key architectural challenges such as: platform interface stability, evolution management, security, reliability, how to support the business strategy, suitable architectures to support open source style development; how open and flexible an architectural is.	5,8,11,21,2 7
Heterogeneity of software licenses and systems evolution in an ecosystem. Organizations must manage these issues in order to decrease risks of dependence.	5,31,41
Companies have difficulty at establishing a set of resources in order to differentiate from competitors. It is necessary a correct engagement of the keystone organization in the social dimension.	3,7
Technical and socio-organizational barriers for coordination and communication of requirements in geographic distributed projects.	16,30
Infrastructure and tools for fostering social interaction, decision-making and development across organizations involved in both open source and proprietary ecosystems.	9,14,17,28

Perspectives and Definitions of Software Ecosystems

Table 1.1 Categories of software ecosystems

Category	Examples	Components	Contributors
digital platforms	mobile app stores, integrated development environments	mobile apps, software plug-ins, or extensions	third-party app or plug-in developers and their users
social coding platforms	SourceForge, GitHub, GitLab, Gitea, Bitbucket	software project repositories	software project contributors
component-based software ecosystems	software library registries (e.g., CRAN, npm, RubyGems, PyPi, Maven Central), OS package registries (e.g., Debian packages, Ubuntu package archive)	interdependent software packages	consumers and producers of software packages and libraries
software automation ecosystems	Docker Hub, Kubernetes, Ansible Galaxy, Chef Supermarket, Puppet Forge	container images, configuration and orchestration scripts, CI/CD pipelines and workflows	creators and maintainers of workflow automation, containerization and orchestration solutions
communication-oriented ecosystems	mailing lists, Stack Overflow, Slack	e-mail threads, questions, answers, messages, posts, etc.	programmers, developers, end users, researchers
OSS communities	Apache Software Foundation, Linux Foundation	OSS projects	community members, code contributors, project maintainers, end users

1. digital platforms

- 一个软件生态系统，其中包括一个平台所有者实施治理机制，以促进数字平台上的价值创造机制，在平台所有者和自主互补者及消费者的生态系统之间。
- 这与Bosch等人先前提到的定义一致

2. component-based software ecosystems

- 包括大量可重复使用的软件组件，这些组件之间通常存在许多相互依赖关系。
- 专注于基于依赖关系的重用技术细节，这使其与数字平台生态系统研究有所区别，后者更加注重商业和管理方面。

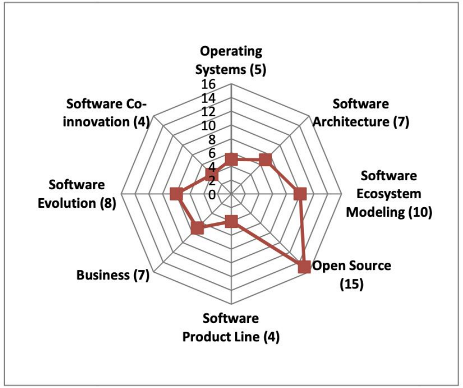


Fig. 3. The most common areas studied in software ecosystems.

Perspectives and Definitions of Software Ecosystems

Table 1.1 Categories of software ecosystems

Category	Examples	Components	Contributors
digital platforms	mobile app stores, integrated development environments	mobile apps, software plug-ins, or extensions	third-party app or plug-in developers and their users
social coding platforms	SourceForge, GitHub, GitLab, Gitea, Bitbucket	software project repositories	software project contributors
component-based software ecosystems	software library registries (e.g., CRAN, npm, RubyGems, PyPi, Maven Central), OS package registries (e.g., Debian packages, Ubuntu package archive)	interdependent software packages	consumers and producers of software packages and libraries
software automation ecosystems	Docker Hub, Kubernetes, Ansible Galaxy, Chef Supermarket, Puppet Forge	container images, configuration and orchestration scripts, CI/CD pipelines and workflows	creators and maintainers of workflow automation, containerization and orchestration solutions
communication-oriented ecosystems	mailing lists, Stack Overflow, Slack	e-mail threads, questions, answers, messages, posts, etc.	programmers, developers, end users, researchers
OSS communities	Apache Software Foundation, Linux Foundation	OSS projects	community members, code contributors, project maintainers, end users

3. social coding platforms

- 在过去的二十年中，基于Web的代码托管平台的格局发生了许多重要变化，如图1.2所示。
- GitHub维护着数亿个开源软件存储库的历史信息，并成为许多实证研究的对象，重点关注不同方面。GitHub是第一个社交编码平台。
- - 围绕GitHub项目的社交方面已经得到了广泛研究，包括沟通模式、通过拉取请求进行合作、贡献者工作量变化、性别和任期多样性、地理分布式开发,项目之间的技术与社会对齐 ,游戏化对软件开发人员行为影响 ,情感分析及情感分析析，项目复刻现象
- 关于GitHub项目中开发活动自动化的研究，比如CI/CD工具使用，开发机器人使用

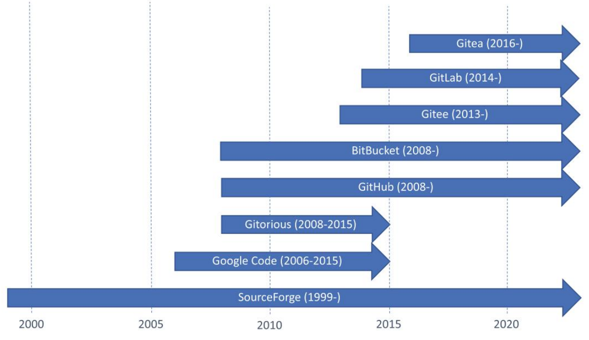


Fig. 1.2 Historical overview of source code hosting platforms

Perspectives and Definitions of Software Ecosystems

Table 1.1 Categories of software ecosystems

Category	Examples	Components	Contributors
digital platforms	mobile app stores, integrated development environments	mobile apps, software plug-ins, or extensions	third-party app or plug-in developers and their users
social coding platforms	SourceForge, GitHub, GitLab, Gitea, Bitbucket	software project repositories	software project contributors
component-based software ecosystems	software library registries (e.g., CRAN, npm, RubyGems, PyPi, Maven Central), OS package registries (e.g., Debian packages, Ubuntu package archive)	interdependent software packages	consumers and producers of software packages and libraries
software automation ecosystems	Docker Hub, Kubernetes, Ansible Galaxy, Chef Supermarket, Puppet Forge	container images, configuration and orchestration scripts, CI/CD pipelines and workflows	creators and maintainers of workflow automation, containerization and orchestration solutions
communication-oriented ecosystems	mailing lists, Stack Overflow, Slack	e-mail threads, questions, answers, messages, posts, etc.	programmers, developers, end users, researchers
OSS communities	Apache Software Foundation, Linux Foundation	OSS projects	community members, code contributors, project maintainers, end users

4. OSS communities

- 这些OSS生态系统相对于封闭的专有软件生态系统具有明显优势。遵循“足够多的眼睛看，所有错误都是浅显”的箴言。
- OSS生态系统受益于潜在非常庞大数量的人可以报告错误、审查代码并识别潜在安全问题。只要所使用的软件许可证兼容，组织和公司就可以通过依赖于OSS组件而节省资金，而不是重新发明轮子并自行开发这些组件。
- OSS生态系统及其构成部分通常是由无偿开发者志愿维护。
- OSS生态系统通常由非营利软件基金会控制、维护和托管。（Apache软件基金会、Linux Foundation、第三方组件的OSS项目社区）
- 存在着许多完全私有和由公司控制的软件生态系统。它们很少成为定量经验研究的主题，但很可能这些私有生态系统具有许多已知开源软件生态系统的特征。（MATLAB、SAP）

Perspectives and Definitions of Software Ecosystems

Table 1.1 Categories of software ecosystems

Category	Examples	Components	Contributors
digital platforms	mobile app stores, integrated development environments	mobile apps, software plug-ins, or extensions	third-party app or plug-in developers and their users
social coding platforms	SourceForge, GitHub, GitLab, Gitea, Bitbucket	software project repositories	software project contributors
component-based software ecosystems	software library registries (e.g., CRAN, npm, RubyGems, PyPi, Maven Central), OS package registries (e.g., Debian packages, Ubuntu package archive)	interdependent software packages	consumers and producers of software packages and libraries
software automation ecosystems	Docker Hub, Kubernetes, Ansible Galaxy, Chef Supermarket, Puppet Forge	container images, configuration and orchestration scripts, CI/CD pipelines and workflows	creators and maintainers of workflow automation, containerization and orchestration solutions
communication-oriented ecosystems	mailing lists, Stack Overflow, Slack	e-mail threads, questions, answers, messages, posts, etc.	programmers, developers, end users, researchers
OSS communities	Apache Software Foundation, Linux Foundation	OSS projects	community members, code contributors, project maintainers, end users

5. Communication-oriented ecosystems

- 面向通信的生态系统，在这些生态系统中，主要组成部分是一些社交通信构件，通过某种通信渠道在软件社区成员之间共享。（邮件列表、开发者讨论论坛、问答平台如Stack Overflow）、以及现代通信平台（如Slack和Discord））
- 它们所包含的主要组成部分（例如问题、答案、帖子、电子邮件和消息线程）大多基于非结构化或半半结构化文本。从中提取和分析相关信息需要基于自然语言处理（NLP）的特定技术。
- 研究人员出于各种原因对这些〝社交程序员生态系统〞进行了分析，大多数情况下是从社会角度出发。例如邮件列表、讨论论坛。

Perspectives and Definitions of Software Ecosystems

Table 1.1 Categories of software ecosystems

Category	Examples	Components	Contributors
digital platforms	mobile app stores, integrated development environments	mobile apps, software plug-ins, or extensions	third-party app or plug-in developers and their users
social coding platforms	SourceForge, GitHub, GitLab, Gitea, Bitbucket	software project repositories	software project contributors
component-based software ecosystems	software library registries (e.g., CRAN, npm, RubyGems, PyPi, Maven Central), OS package registries (e.g., Debian packages, Ubuntu package archive)	interdependent software packages	consumers and producers of software packages and libraries
software automation ecosystems	Docker Hub, Kubernetes, Ansible Galaxy, Chef Supermarket, Puppet Forge	container images, configuration and orchestration scripts, CI/CD pipelines and workflows	creators and maintainers of workflow automation, containerization and orchestration solutions
communication-oriented ecosystems	mailing lists, Stack Overflow, Slack	e-mail threads, questions, answers, messages, posts, etc.	programmers, developers, end users, researchers
OSS communities	Apache Software Foundation, Linux Foundation	OSS projects	community members, code contributors, project maintainers, end users

6. software automation ecosystems

- 基于云的平台自动化部分管理、开发、打包、部署、交付、配置和编排软件应用程序的技术解决方案：容器化解决方案，基于基础设施即代码的编排工具，以及用于自动化DevOps和CI/CD的工具。
- 容器化允许开发人员将其应用程序的所有（软件和数据）组件打包到所谓的容器中，这些容器是轻量级、便携且自包含的可执行软件环境，可以在任何操作系统或云平台上运行。
- 通过基础设施即代码（IaC）进行基础设施管理，基础设施管理工具可以实现自动化的机器配置、部署、扩展和负载均衡。（Ansible Galaxy、Chef Supermarket和Puppet Forge）
- DevOps和CI/CD协作式分布式软件开发流程，特别是在社交编码平台上托管的软件项目，往往会使用持续集成、部署和交付工具（CI/CD）进行优化和自动化，这些工具是DevOps实践的关键组成部分。
- CI/CD工具使项目维护者能够指定项目特定的工作流程或流水线，从而自动化许多重复性高且容易出错的人类活动，这些活动是开发过程中的一部分。
- 目前有专注于围绕GitHub Actions形形成的生态系统，即GitHub集成CI/CD服务。

Data Sources for Mining Software Ecosystem

挖掘软件仓库（MSR）研究社区依赖于各种公开可访问的原始数据、API或其他数据提取工具、数据转储、策划好的数据集以及数据处理工具（例如专用解析器），这取决于正在进行的研究的特定目的和需求。

1. 优势

这些数据源及其相关工具为软件工程领域的实证研究提供了丰富的资源。

2. 缺点

- 现有的数据和工具很快就会过时，因为要跟上原始数据源或访问它们所需的API的变化是困难且需要大量精力。许多创建和维护数据提取工具或策划数据集的倡议已经停止，主要是由于缺乏持续资金支持或因为最初的维护者由于职业变动而放弃了该倡议。
- 由于道德、法律或隐私原因，可能会阻止感兴趣的数据的特定部分被公开。
- 特定分析可能需要特定类型的数据，这些数据在现有数据集中并不容易获得，需要创建新的数据集或扩展现有数据集。
- 因为数据的收集或过滤方式，现有数据集可能不适合特定的分析。
- 依赖原始数据源而非经过精心筛选的数据集进行研究可能会降低可重复性，因为与已发布的数据集不同，在发表研究结果后无法保证原始数据仍然保持不变。

Data source

1. GitHub

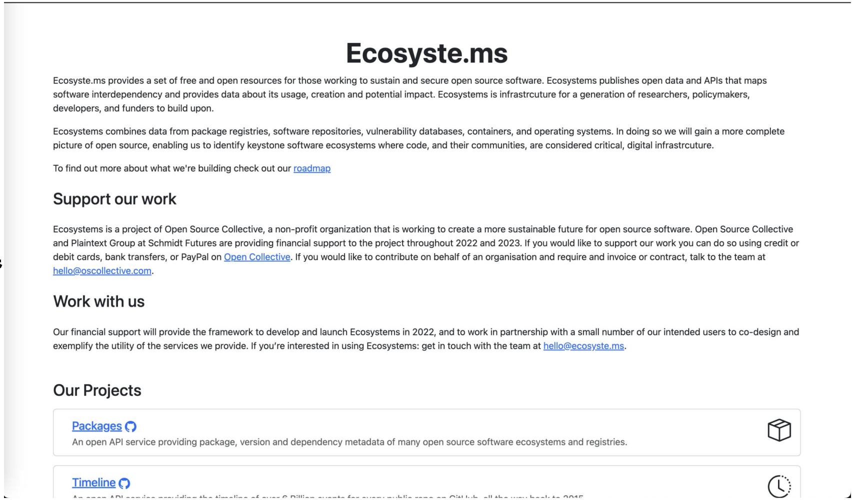
- GitHub提供公共REST和GraphQL API以与其庞大的事件数据集和与托管存储库的互动进行交互。
- GHArchive
- GHTorrent（2021年3月份停止维护）
- TravisTorrent

2. JAVA

- Qualitas Corpus（2010，2013）
- Apache的Maven中央仓库
- Maven Dependency Dataset

3. 软件库（Software Library）

- Libraries.io （2017-2020，5次存储，已经停止维护）
- Ecosyste.ms project（2022-， Open Collective 赞
  - 来自数十个软件包注册表（编程语言和Linux发行版）的数百万开源软件包的索引，每天都会添加数以万计的新软件包版本。
  - 源自多种不同来源的数千万开源仓库和Docker项目及其依赖项的索引，每天都会添加成千上万个新仓库
  - 提供软件存储库、贡献者和安全漏洞元数据服务范围广泛，解析软件依赖关系和许可证元数据，解决软件包依赖树问题，并生成程序发布之间的差异等。



Data source

4. 其他的数据源

- Software Heritage ecosystem（最大的公共软件归档，联合国教科文组织UNESCO支持）
- World of Code（WoC）（提供了工具便于分析技术依赖关系、社交网络及其相互关系）
- Boa（提供了领域特定语言和分布式计算基础设施）https://boa.cs.iastate.edu/

5. 停止维护的数据源

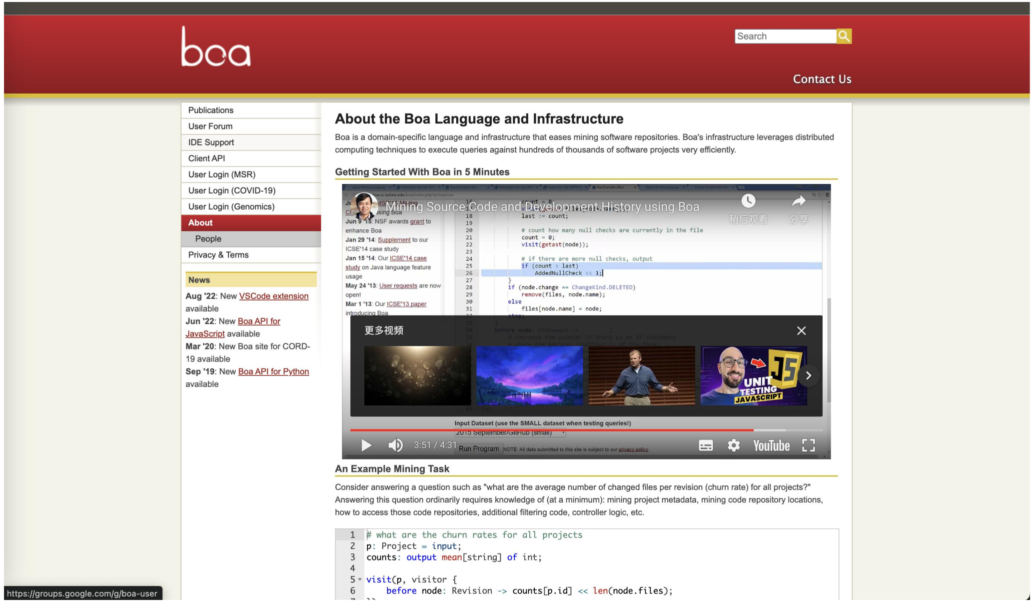
- PROMISE Software Engineering Repository
- FLOSSmole
- Candoia
- Sourcerer
- DebSources
- Jira

6. The CHAOSS Project

- Augur
- Grimoire-Lab

7. OpenDigger

- Metrics
- OSSGraph
- Open- LeaderBoard
- Open-Perf



总结

本次报告为软件生态系统研究的新手提供了入门指南，帮助他们踏出第一步。报告的目的是提供关键资料，使人们能够迅速了解该领域的基本概念和发展
内容包括

- 回顾了软件生态系统的起源，梳理了该领域的历史发展。
- 强调了从不同视角对软件生态系统的理解及其相关定义。
- 对不同类型的软件生态系统进行了分类梳理。
- 提供了分析开源软件生态系统的关键工具，帮助读者在实践中运用。

谢谢

THANK YOU FOR WATCHING