

Produire de l'open source

Pourquoi et comment créer une communauté ?

Stéfane Fermigier

Cours à l'EPITA - 18 oct. 2022

Une communauté: pourquoi?

Out-spend or **Out-teach**

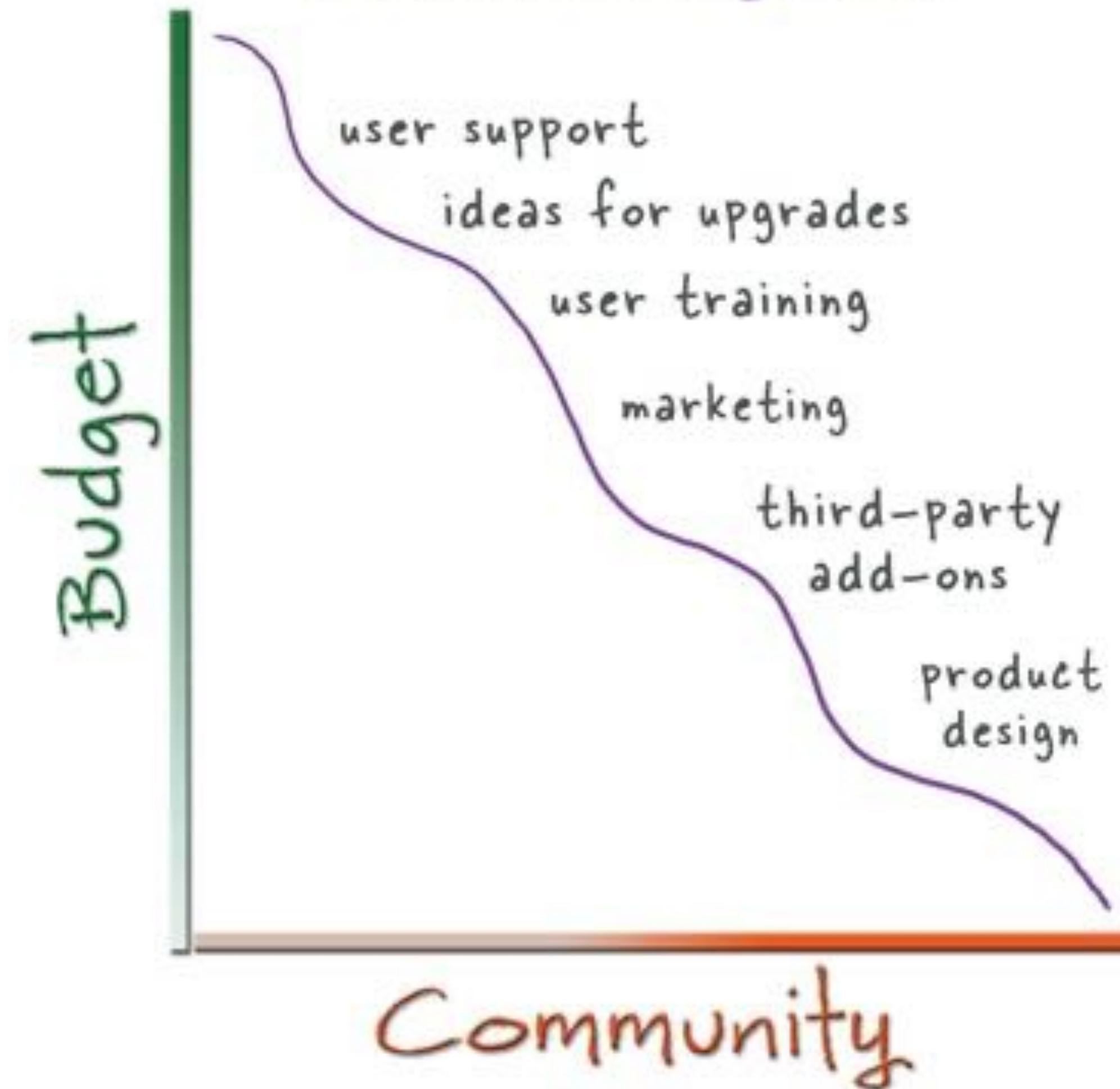


Source: Kathy Sierra



Source: Kathy Sierra

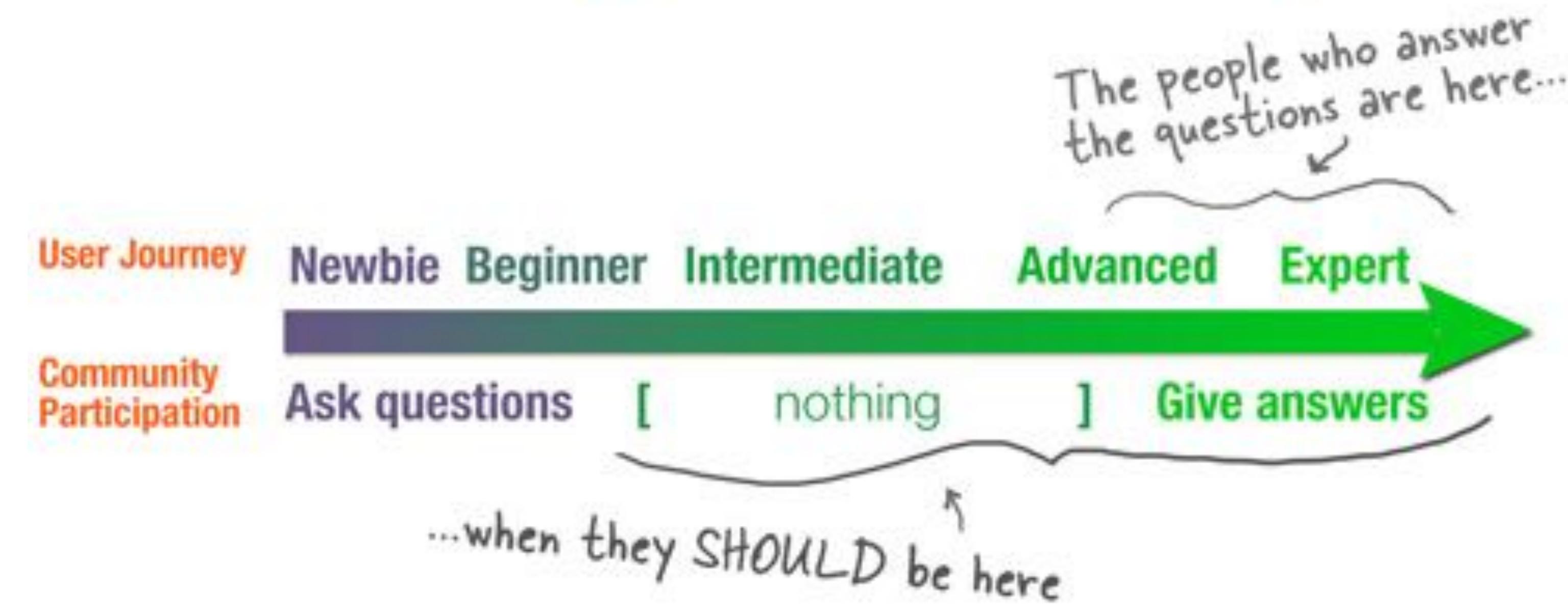
Community ROI



the more your users participate,
the less money you need

Source: Kathy Sierra

Building a User Community



The earlier they start helping others, the faster the community grows, and the more likely it is that not-yet-advanced users stick with it

Source: Kathy Sierra

→ Common goals

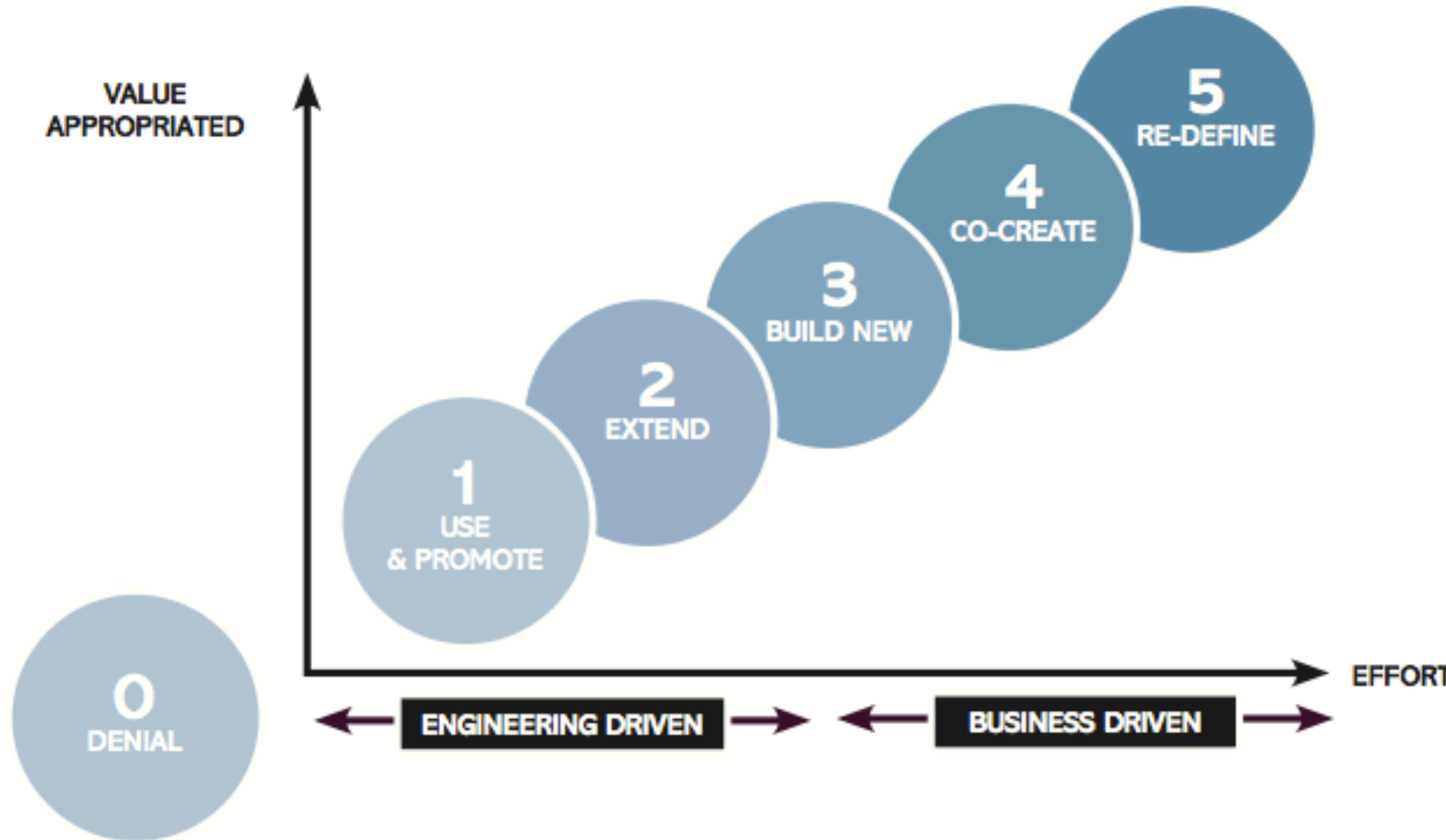
- Get **feedback**
- Get **contributors**
- Improve our software **quality**
- Generate **buzz** and **evangelists**
- **Show** that we do have a community

Stratégie open source

Evolution classique

- *Software developed by communities of individuals*
- *Vendors begin to engage with existing open source communities*
- *Vendor-dominated open source development and distribution projects*
- *Corporate-dominated open source development communities*

Modèle d'adoption



Source: Eclipse Foundation/Carleton University & Nortel

Eléments de stratégie - Pour un éditeur open source

- *Software License*
- *Copyright Ownership*
- *Development Model / Community*
- *Revenue Generator*

Marketing et “évangélisation”

Site Web

- Design
 - Utiliser / acheter un template “pro”
 - Tendance récente: Twitter Bootstrap
- Pitch (5 lignes)
 - Doit parler à des non-spécialistes
- *Features / benefits*

Site Web

- Définir l'audience cible
 - Segmenter si nécessaire
 - *Progressive disclosure*
 - 1 minute / 5 minutes / 1 heure
 - News et roadmap
 - Montrer qu'il y a de l'activité

Site Web

- Liens vers les outils communautaires (*cf. infra*)
- Liens vers les resources documentaires
 - Doc (architecture, utilisateurs)
 - Slides (SlideShare ou SpeakerDeck)
 - Screencasts

Le code

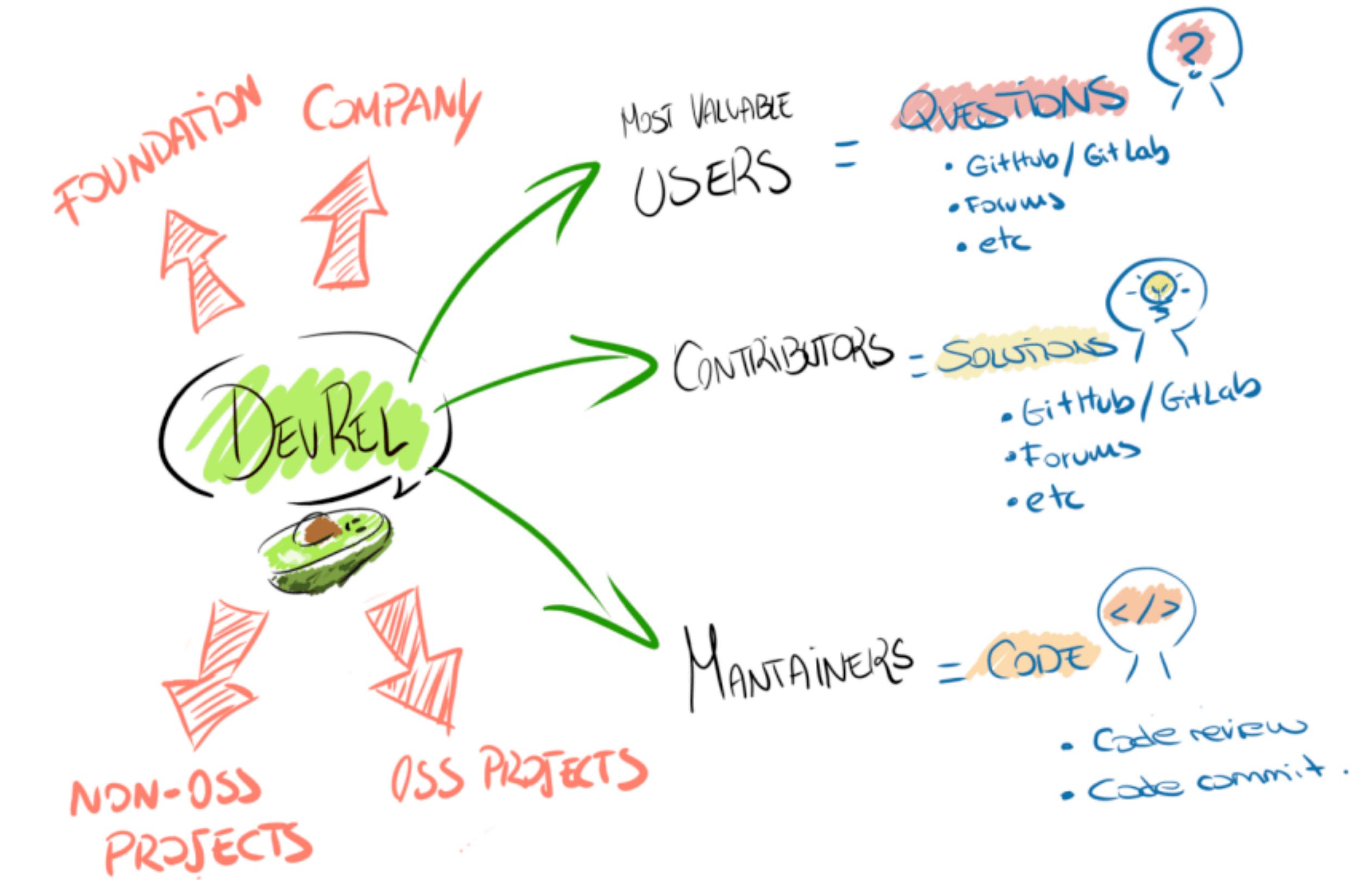
- Doit être facile à trouver, à builder (“configure ; make ; make install”)
 - Comment gérer les dépendances ?
- README, INSTALL, etc.
 - Note: le fichier README est devenu crucial avec des outils comme GitHub
- Packaging (distrib Linux, Mac, Win...)

L'évangélisation selon Guy Kawasaki

- “Evangelism” became a business buzzword during the internet boom of the late 1990s. In fact, as Apple’s second software evangelist, I helped popularize the term. The idea is simple: Derived from a Greek word that means, roughly, “to proclaim good news,” evangelism is explaining to the world how your product or service can improve people’s lives.
- My job at Apple was to proclaim the good news that Macintosh would make everyone more creative and productive. I wasn’t just marketing a computer; I believed in it so much that I wanted others to experience it too. Now, as the chief evangelist of Canva, my job is to share a platform that democratizes design. Evangelists truly have the best interests of others at heart.

Emergence des “DevRel” (developer relations)

“DevRel is the marketing technique used to ensure that one's company, products, and developers establish a good, continuous relationship with external developers through mutual communication.”



Source: <https://blog.bitergia.com/2019/05/28/kpis-and-metrics-for-devrel-programs/>

Activités en lien avec le DevRel

- A DevRel program may comprise a framework built around some or all of the following aspects:
 - **Developer Marketing:** Outreach and engagement activities to create awareness and convert developers to use a product.
 - **Developer Education:** Product documentation and education resources to aid learning and build affinity with a product and community.
 - **Developer Experience (DX):** Resources like a developer portal, product, and documentation, to activate the developer with the least friction.
 - **Developer Success:** Activities to nurture and retain developers as they build and scale with a product.
 - **Community:** Nourishes a community to maintain a sustainable program.

Exemples d'animation

- Participation à / organisation de conférences
- Workshops
- Sprints
- Hackathons
- Club utilisateurs

DevRel vs. Community management

- La dénomination “Community manager” est trompeuse, le plus souvent il s’agit de “correspondants sur les réseaux sociaux”

Gouvernance et modèle de développement

Modèles de gouvernance

- *Vendor-led*
 - Concessions possibles: club utilisateur, *board* plus ou moins indépendant et influent
- *Community led*
 - Formel ou informel
 - Communauté établie (“Fondation”: FSF, ASF, Eclipse, OW2...) ou ad-hoc

Good Governance Initiative - OW2 & OSPO.zone



Abraham Maslow's Hierarchy of Behavioral Motivation



OW2 Goals to OSS Good Governance

Modèles de décision dans les gouvernance communautaires

- Hiérarchie des membres
 - *Contributeur, committer, core committer...*
- Unanimité, consensus ou *BDFL* ?
 - Qui porte la vision ? Comment est-elle partagée ?
 - Enjeux? Vitesse d'exécution, masse critique ?

Considérations juridiques

Propriété du code

- Centralisée?
 - Chez l'éditeur
 - Au sein d'une communauté
- Ou partagée?
- Notion de *contributor's agreement* (CLA / CCA) + DCO

Choix des licences

- Contrat moral avec la communauté
 - Tout changement risque d'être vécu de manière traumatique
- Contraintes business
 - Ex: *open core, double licensing*
 - Copyleft / weak copyleft / pas de copyleft

Choix des licences

- ~100 licences reconnues par l'OSI, 8 “*popular and widely used or with strong communities*”
 - BSD, MIT, (L)GPL, APL, MPL, EPL, CDDL
- Critères importants:
 - Compatibilité GPL (en général désirable)
 - Compatibilité intégration avec du propriétaire (choix)

Évolution des pratiques

FOSS 0.1
1983-1990



Richard Stallman, Founder of the Free Software Movement

Photo credit: <http://commons.wikimedia.org/wiki/User:Vicapowell39>

- The free software movement was started in 1983 by Richard Stallman
- Most of the open source software produced at the time was developed by **very small teams** (2-3 persons), using local development tools
- Software were distributed using **tapes**, then **FTP**
- Marketing was mostly through word-of-mouth
- Community = people with internet access (not many people)

Early successes

- The GNU “operating system” (minus the kernel) was already displacing proprietary tools in the early 90s
- The moral and legal frameworks upon which the free software (and later, the open source) movement is built
 - Didn’t mandate / prescribe any production model for free software, though

Challenges

- Economic and moral questioning:
 - **Is it ok to make money with free software?**
 - How to make the system **sustainable**?
 - How to **scale development** efforts to larger teams?

FOSS 0.9
1991-1998

Successes

- **Larger scale projects** start to appear, attracting tens, then hundreds of developers (and later, thousands)
- **Tools and practices** are developed, most often on top of existing internet protocols to address the needs of distributed development at this scale :
 - Centralized source code management
 - Mailing lists or usenet forums

Successes

- Linux (1991)
- The Debian (1993) and Red Hat (1994) distributions
- The Apache Web Server (1995)

FOSS I.O
1998-2007

- **Open source** becomes the preferred term for most free software based businesses
- The **Web** becomes pervasive
- Several organizations created to foster **governance** of open source projects (Apache Foundation, Eclipse Foundation, OW2...)
- Several successful IPOs on top of the Web 1.0 bubble (Red Hat, VA Linux), Netscape open sources the Mozilla browser...

The 4 engines of collaboration

- Real-time **shared vision**
- Real-time **status updates**
- Real-time **help requests**
- Self-service **archives**

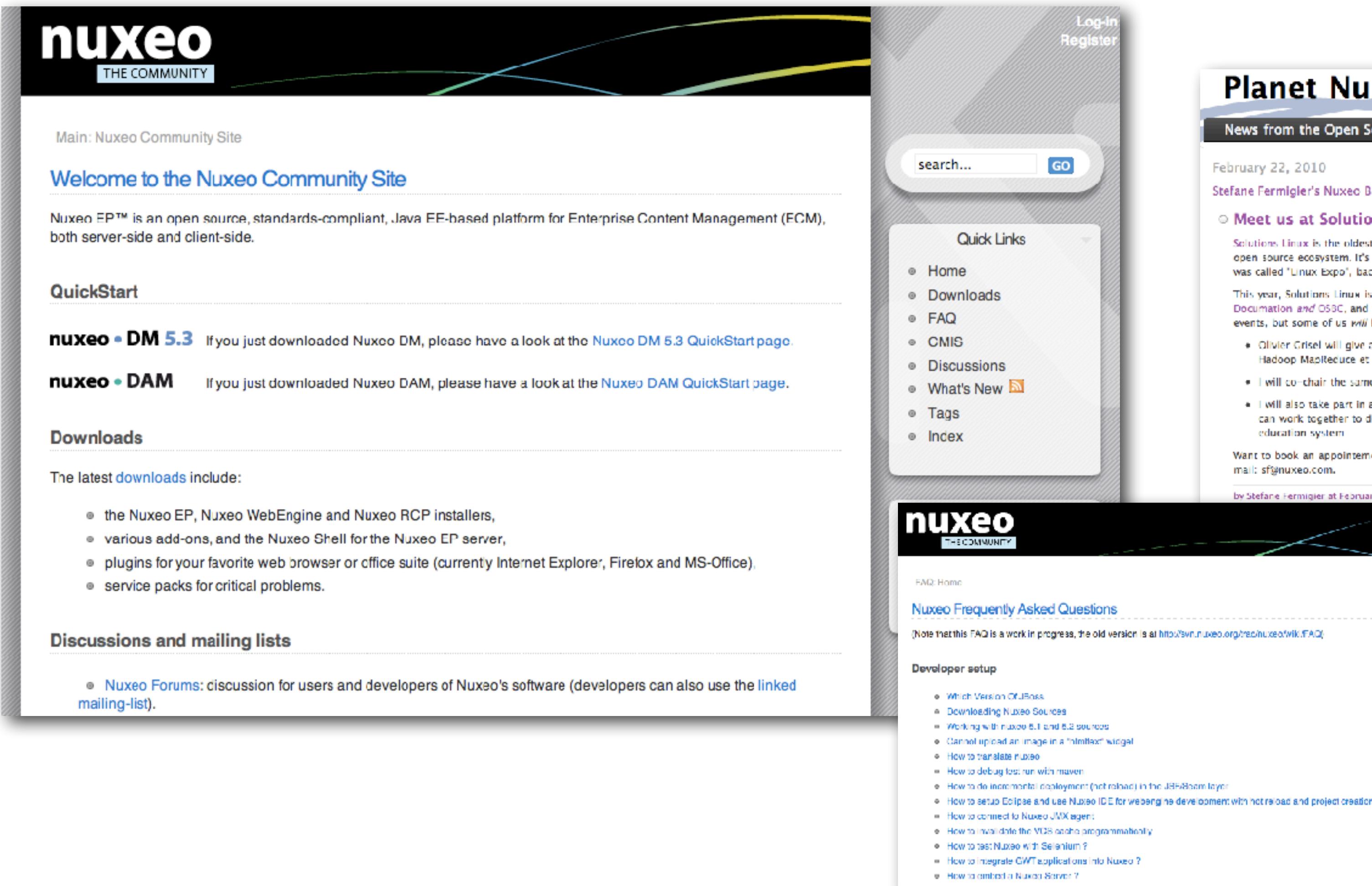
Source: Bertrand Delacretaz, 2009

OS Secret Sauce

“Every successful open source project I know uses **PRIM**. Every closed source project I know, doesn't. People wonder how open source projects manage to create high-quality products without managers or accountability. The answer: **we're accountable to our infrastructure**. PRIM is the **open source secret sauce**.”

Ted Husted <http://jroller.com/TedHusted/entry/prim>

P = Portal (often, a Wiki)



The screenshot shows the Nuxeo Community Site homepage. At the top, there's a dark header with the "nuxeo" logo and "THE COMMUNITY". Below the header, the main content area has a black background with a wavy pattern. It features a "Welcome to the Nuxeo Community Site" section, a "QuickStart" section for Nuxeo DM 5.3, a "Downloads" section listing recent releases, and a "Discussions and mailing lists" section. On the right side, there's a sidebar with a "Quick Links" menu containing links to Home, Downloads, FAQ, CMIS, Discussions, What's New, Tags, and Index. A search bar is located at the top right of the sidebar. The footer contains a "FAQ: Home" link and a "Nuxeo Frequently Asked Questions" section.



The screenshot shows the "Planet Nuxeo" news page. The header includes the "nuxeo" logo and "THE COMMUNITY". The main content area features a news item by Stefane Fermigier about Solutions Linux in March. The sidebar on the right is titled "Don't miss a thing!" and lists several blog entries from different authors. It also includes sections for "Our blogs" (with links to various Nuxeo blogs), "About Us" (describing Nuxeo as a friendly open-source vendor), and social media links for Twitter, LinkedIn, and the Nuxeo website. The footer contains links for "DOWNLOAD NUXEO • DM", "ALL NUXEO • DM SCREENCASTS", and "DISCOVER OUR PRODUCTS".

R = Repository

Repositories list					Mercurial
Name	Description	Contact	Last change		
addons	Nuxeo Addons	Florent Guillaume <fg@nuxeo.com>	2 weeks ago	zip	RSS Atom
addons/nuxeo-core-folder-versioning	Nuxeo Core Folder Versioning	Florent Guillaume <fg@nuxeo.com>	12 months ago	zip	RSS Atom
addons/nuxeo-correspondence-marianne	Nuxeo Correspondance Mariane	Florent Guillaume <fg@nuxeo.com>	11 minutes ago	zip	RSS Atom
addons/nuxeo-features-page-layout	Nuxeo Features Page Layout	Florent Guillaume <fg@nuxeo.com>	9 months ago	zip	RSS Atom
addons/nuxeo-generic-wss-handler	Nuxeo Generic WSS Handler	Florent Guillaume <fg@nuxeo.com>	4 months ago	zip	RSS Atom
addons/nuxeo-http-client	Nuxeo HTTP Client	Florent Guillaume <fg@nuxeo.com>	4 weeks ago	zip	RSS Atom
addons/nuxeo-platform-annotations	Nuxeo Platform Annotation service addon	Florent Guillaume <fg@nuxeo.com>	3 months ago	zip	RSS Atom
addons/nuxeo-platform-archive	Nuxeo Platform Archive service addon	Florent Guillaume <fg@nuxeo.com>	10 months ago	zip	RSS Atom
addons/nuxeo-platform-archive-importer	Nuxeo Platform Archive Importer plugin	Florent Guillaume <fg@nuxeo.com>	18 months ago	zip	RSS Atom
addons/nuxeo-platform-audio	Nuxeo Platform Audio Addon	Florent Guillaume <fg@nuxeo.com>	3 weeks ago	zip	RSS Atom
addons/nuxeo-platform-audit-web-access	Nuxeo Platform Audit Web Access	Florent Guillaume <fg@nuxeo.com>	3 months ago	zip	RSS Atom
addons/nuxeo-platform-categorization	Nuxeo Automated Document Categorization: framework and service	Florent Guillaume <fg@nuxeo.com>	2 weeks ago	zip	RSS Atom
addons/nuxeo-platform-categorization/nuxeo-platform-categorization-coverage	Nuxeo Automated Document Categorization: dc:coverage model	Florent Guillaume <fg@nuxeo.com>	2 weeks ago	zip	RSS Atom
addons/nuxeo-platform-categorization/nuxeo-platform-categorization-language	Nuxeo Automated Document Categorization: dc:language model	Florent Guillaume <fg@nuxeo.com>	2 weeks ago	zip	RSS Atom
addons/nuxeo-platform-categorization/nuxeo-platform-categorization-subjects	Nuxeo Automated Document Categorization: dc:subjects model	Florent Guillaume <fg@nuxeo.com>	2 weeks ago	zip	RSS Atom
addons/nuxeo-platform-document-categorization	Nuxeo Automated Document Categorization	Florent Guillaume <fg@nuxeo.com>	2 weeks ago	zip	RSS Atom
addons/nuxeo-platform-documentLink	Application Level document proxies for Nuxeo5	Florent Guillaume <fg@nuxeo.com>	4 months ago	zip	RSS Atom

I = Issue (or Bug)Tracker

The image shows two side-by-side screenshots of issue tracking systems.

Bugzilla Screenshot: A Mozilla Firefox browser window displaying a Bugzilla bug detail page. The URL is https://bugzilla.mozilla.org/show_bug.cgi?id=169837. The bug details include:

- Bug #:** 169837
- Product:** Firefox
- Component:** Bookmarks
- Status:** NEW
- Resolution:** (empty)
- Assigned To:** Vladimir Vukicevic (Bookmarks Bugs Only) <vladimir+bm@vld1.com>
- QA Contact:** mconnor@steelgriffin.com
- URL:** (empty)
- Summary:** scroll bookmarks but not other menu items in bookmarks menu
- Status Whiteboard:** (empty)
- Keywords:** (empty)
- Attachment:** patch (2004-07-25 08:55 PDT, 5.74 KB, new, Edit | Diff)
- Depends on:** (empty)
- Blocks:** (empty)
- Votes:** 8
- Additional Comments:** (empty)

Nuxeo Screenshot: A screenshot of the Nuxeo Open Source ECM interface. The top navigation bar includes links for HOME, BROWSE PROJECT, FIND ISSUES, CREATE NEW ISSUE, ADMINISTRATION, and QUICK SEARCH. The main content area displays the following information:

- Project Lead:** Thierry Delprat
- URL:** <http://www.nuxeo.org/>
- Actions:**
 - Create a new issue in project Nuxeo Enterprise Platform
 - Administer Project
 - Release Notes
- Select:** Open Issues, Road Map, Change Log, Popular Issues, Labels, Versions, Components
- Components:** Audit, Comments, Core, Core JCR Storage, Core SQL Storage, Desktop Integration, Directory, Flex, Installers / Admin tools, Notifications, Packaging, Relation Engine, Runtime, Search / Query, Security, Tagging, Transforms, Web Foundations, Web UI.
- Versions:** 5.1.7, 5.2 GA SP1, 5.2.1.0, 5.3 GA, 5.3.1, 5.3.2, 5.3.x, 6.0, Unscheduled.
- Reports:** Recently Created Issues Report, Created vs Resolved Issues Report, Resolution Time Report, Average Age Report, Pie Chart Report, SUPNXP issues over time, SUPNXP issues by projects, User Workload Report, Version Workload Report, Time Tracking Report, Single Level Group By Report.
- Preset Filters:** All, Outstanding, Unscheduled, Assigned to me, Reported by me, Resolved recently, Added recently, Updated recently, Most important.
- Project Summary:** Open: 186 (4%), In Progress: 11, Reopened: 12, Resolved: 2783 (61%), Closed: 1153 (25%), Triage: 427 (9%).
- Open Issues:** By Priority: Blocker: 6 (1%), Critical: 20 (3%).

M = Mailing List (+ foruM)

All Mailboxes (Found 10431 matches for search)

From	Date received	Mailbox
jiveforums-ecm...	Feb 19, 2010 1:27 PM	Lists
[Ecm] LAN Web Access	Feb 19, 2010 3:24 PM	Lists
Re: [Ecm] How can I configure a multi...	Feb 19, 2010 3:26 PM	Lists
Re: [Ecm] LAN Web Access	Feb 19, 2010 3:29 PM	Lists
Re: [Ecm] LAN Web Access	Feb 19, 2010 5:33 PM	Lists
[Ecm] Errors starting Nuxeo 5.3.boss	Feb 19, 2010 8:29 PM	Lists
[Ecm] Sharepoint WSS behind Apache...	Feb 20, 2010 2:19 PM	Lists
Re: [Ecm] Sharepoint WSS behind Ap...	Feb 20, 2010 4:51 PM	Lists
[Ecm] Errors starting Nuxeo 5.3 ...	Feb 20, 2010 4:54 PM	Lists
Re: [Ecm] Gestion des courriers	Feb 22, 2010 9:59 AM	Lists
jiveforums-ecm...	Feb 22, 2010 2:48 PM	Lists
jiveforums-ecm...	Feb 22, 2010 2:50 PM	Lists
Maxime Alarie	Feb 22, 2010 6:12 PM	Lists
Daniel Tellez	Feb 22, 2010 7:10 PM	Lists
Re: [Ecm] looking for Nuxeo DM to...	Feb 22, 2010 8:12 PM	Lists

Nuxeo mailing list archives

February 2010 Archives by date

- Messages sorted by: Thread Subject Author
- More info on this list...

Starting: Mon Feb 1 08:46:58 CET 2010
Ending: Thu Feb 25 21:04:59 CET 2010
Messages: 200

- [Ecm] Templates with predefined workflows for documents jiveforums-ecm at nuxeo.com
- [Ecm] Reassign delete part2 Radu Andrei Tenescu
- [Ecm] navigation jiveforums-ecm at nuxeo.com
- [Ecm] recursive delete part2 Radu Andrei Tenescu
- [Ecm] Migration de 5.1.4 vers 5.3 jiveforums-ecm at nuxeo.com
- [Ecm] Nuxeo and WebDAV Maxime Alarie
- [Ecm] Nuxeo and WebDAV Denizhan YILMAZ
- [Ecm] Migration de 5.1.4 vers 5.3 jiveforums-ecm at nuxeo.com
- [Ecm] website problem jiveforums-ecm at nuxeo.com
- [Ecm] NPE on long-running server Sean Redford
- [Ecm] NPE on long-running server Thierry Delprat
- [Ecm] Nuxeo DM 5.3.0 jiveforums-ecm at nuxeo.com
- [Ecm] Nuxeo DM 5.3.0 jiveforums-ecm at nuxeo.com
- [Ecm] JBoss Seam for Nuxeo DM tomcat + Oracle 10g configuration jiveforums-ecm at nuxeo.com
- [Ecm] Nuxeo DM 5.3.0 jiveforums-ecm at nuxeo.com
- [Ecm] Nuxeo can be integrated in liferay jiveforums-ecm at nuxeo.com
- [Ecm] looking for Nuxeo DM tomcat + Oracle 10g configuration jiveforums-ecm at nuxeo.com
- [Ecm] jiveforums-ecm at nuxeo.com
- [Ecm] issue #24 Feb 22:36 Sean Redford
- commented in lif 25 Feb 14:04 Anelide Tchertchan
- [Ecm] issue #24 Feb 14:38 Bruce Grant
- [Ecm] issue #24 Feb 14:54 Thierry Delprat
- [Ecm] issue #24 Feb 15:04 Bruce Grant
- [Ecm] issue #24 Feb 18:05 Thierry Delprat
- [Ecm] issue #24 Feb 14:22 Bruce Grant
- [Ecm] issue #24 Feb 18:40 Bruce Grant
- [Ecm] issue #24 Feb 22:57 Jesus Ibarra Calvo
- [Ecm] issue #24 Feb 18:12 Maxime Alarie

About

This page is from a mailing archive for one of the Nuxeo forums.

DOWNLOAD NUXEO • DM

ALL NUXEO • DM SCREENCASTS

DISCOVER OUR PRODUCTS

Frow your blogs

Meet us at Solutions Linux in March, 100, by Stefane Fermigier

Nuxeo on the Road (and On the Web), by CheryMcKinnon

See also: Nuxeo EP forums on Web

Automatic prefix to document names

Automatic prefix to document names

AbstractSecurityPolicy

AbstractSecurityPolicy

AbstractSecurityPolicy

AbstractSecurityPolicy

AbstractSecurityPolicy [SOLVED]

AbstractSecurityPolicy

Single sign on trough Nuxeo to php application

Webdav configuration

From: Sean Radford <sean.radford@...>
Subject: Automatic prefix to document names
Newsgroups: gmane.comp.cms.nuxeo.general
Date: 2010-02-24 21:36:46 GMT (16 hours and 16 minutes ago)

Hi,

For some document types we have a requirement to automatically prefix the usual generated document name with an additional node:

i.e. instead of the document with title "My Test Document" getting the name "my-test-document", it would be something like "xyz123-my-test-document"

I first looked into using an event handling listening to the aboutToCreate event, but this does not work as any changes to the document path of the passed-in DocumentModel are not reflected/persisted. (Cannot remember the details but the handling class keeps a reference to the original name/path and uses that instead of the name/path that comes back in the DocumentModel passed in the Event).

As a fall-back I decided to override DocumentActionsBean with my cxa class (higher Seam install precedence) and modified the saveDocument(DocumentModel) appropriately.

Search Google

CVS Client and Browser

Download Now! Windows, Linux, OSX CVS clients for multiple OSes www.aquablob.com

Software Forges, a more integrated approach

- **Sourceforge**, launched in 1999 by VA Linux, integrates all these tools in a consistent Web (1.0) portal
- Makes it super easy for anyone (3.4 million users currently) to start a new open source project (324 000 as of today)
- Several similar products launched afterwards (Collabnet, Trac, Redmine)

Works for non open source software too...

Open Collaboration within Corporations Using Software Forges

**Dirk Riehle, John Ellenberger, Tamir Menahem, Boris Mikhailovski,
Yuri Natchetoi, Barak Naveh, and Thomas Odenwald, SAP**

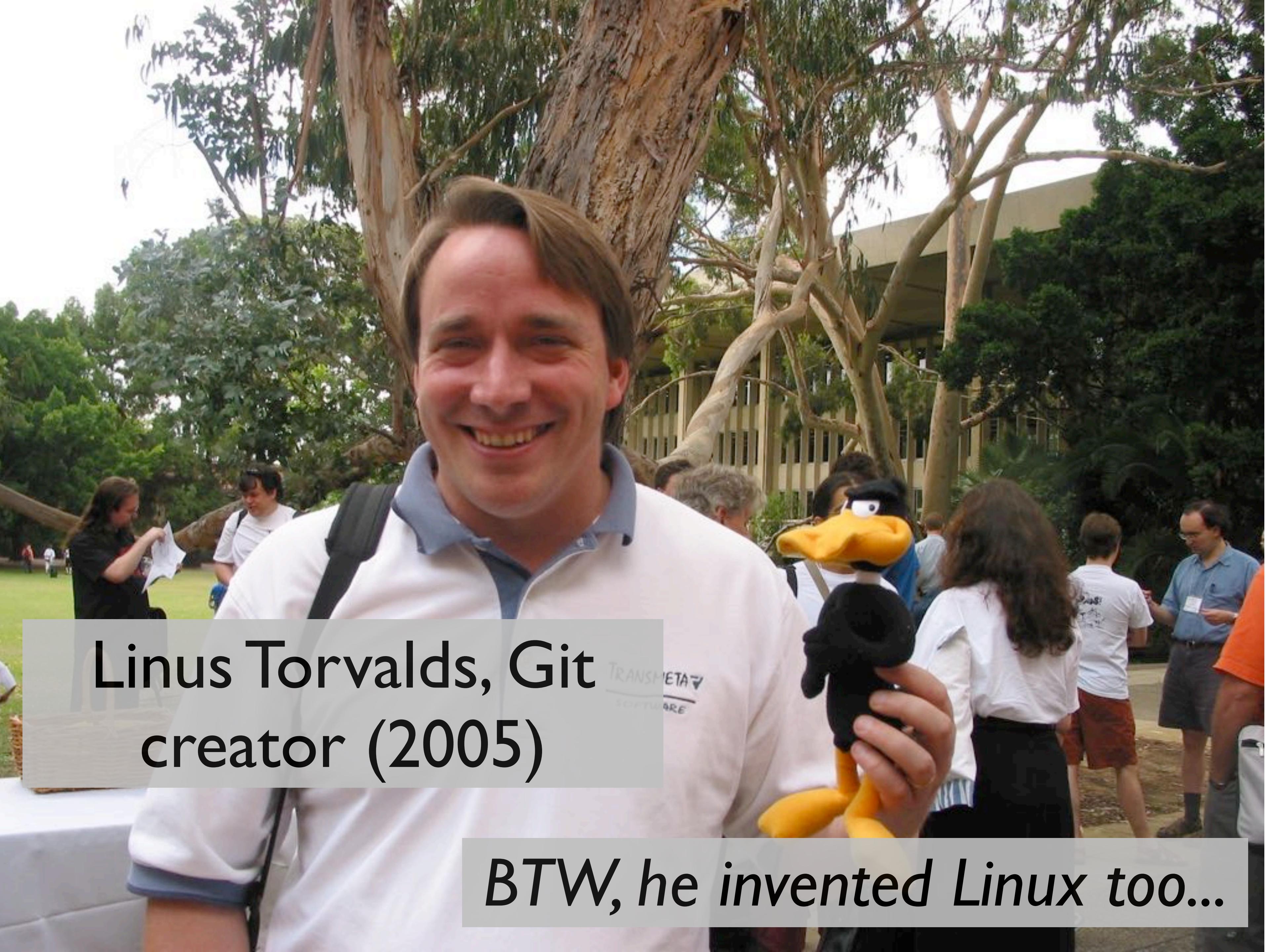


FOSS 2.0
2008-2018

Web 2.0

- Wikipedia (2001)
- Tim O'Reilly's *Architecture of Participation* (2004) and *Web 2.0* (also 2004)
- Consumer Web 2.0, then Enterprise 2.0 replace older applications

- **Git**, and a bunch of other Distributed Source Control Management Systems (DSCM), appear circa 2005 to address the need of very large distributed development teams (1000s of developers for Linux)
- They allow for completely **decentralized development**, and make it much easier for developers to try out new ideas on their own, then “merge” the changes with the main development lines



Linus Torvalds, Git
creator (2005)

BTW, he invented Linux too...

- A new breed of SaaS offerings for developers, such as GitHub (2008) or StackOverflow (2008), appear, leveraging many of the characteristic features of W2.0 or E2.0 applications:
 - Activity streams
 - Social networking
 - Tagging / folksonomies
 - Votes, reputation



Stefane Fermigier

sfermigier

 Nuxeo

 Paris, France

 sf@fermigier.com

 <http://fermigier.com/>

 Joined on May 08, 2010

7
followers

25
starred

0
following

 Contributions

 Repositories

 Public Activity

 Edit Your Profile



a month ago

sfermigier opened issue [mitsuhiko/flask#682](#)

Several issues with the mailing list (librelist), time to look for an alternative?



sfermigier starred [nuxeo/nuxeo-drive](#) a month ago



a month ago

sfermigier pushed to master at [sfermigier/yaka-core](#)

 e572174 Bump Flask-Assets version to 0.8.



a month ago

sfermigier pushed to master at [sfermigier/lemooc-work...](#)

 03af59b Clean up.



sfermigier created branch  master at [sfermigier/lemooc-workshop](#) a month ago



sfermigier created repository [lemooc-workshop](#) a month ago



a month ago

sfermigier pushed to master at [sfermigier/yaka-core](#)

 e8a015f Fix issue with timezones.

GitHub, like SourceForge, but more social



Top Questions

interesting

397

featured

hot

week

month

- 0 0 1 [PHP Curl login into APSX page, after need to post data to Login page](#)
votes answers view [php](#) [asp.net](#) [post](#) [curl](#) [double](#) 20s ago [Rafael Andrade Lott](#) 1

- 0 0 1 [Mass IllegalMonitorStateException on thread wait](#)
votes answers view [java](#) [multithreading](#) [wait](#) [runnable](#) [illegalstateexception](#) 21s ago [JohnM](#) 8

- 0 0 7 [How to add a symlinked JAR to an Eclipse project?](#)
votes answers views [eclipse](#) 22s ago [Robottinosino](#) 1,411

- 1 1 12 [filter a lot of sensitive code from an svn repository](#)
vote answer views [svn](#) [version-control](#) [history](#) [obliterate](#) 37s ago [bahrep](#) 3,410

- 0 0 2 [connect to a access database in java using udl file](#)
votes answers views [java](#) [database](#) [ms-access](#) 45s ago [nigthfox79](#) 47

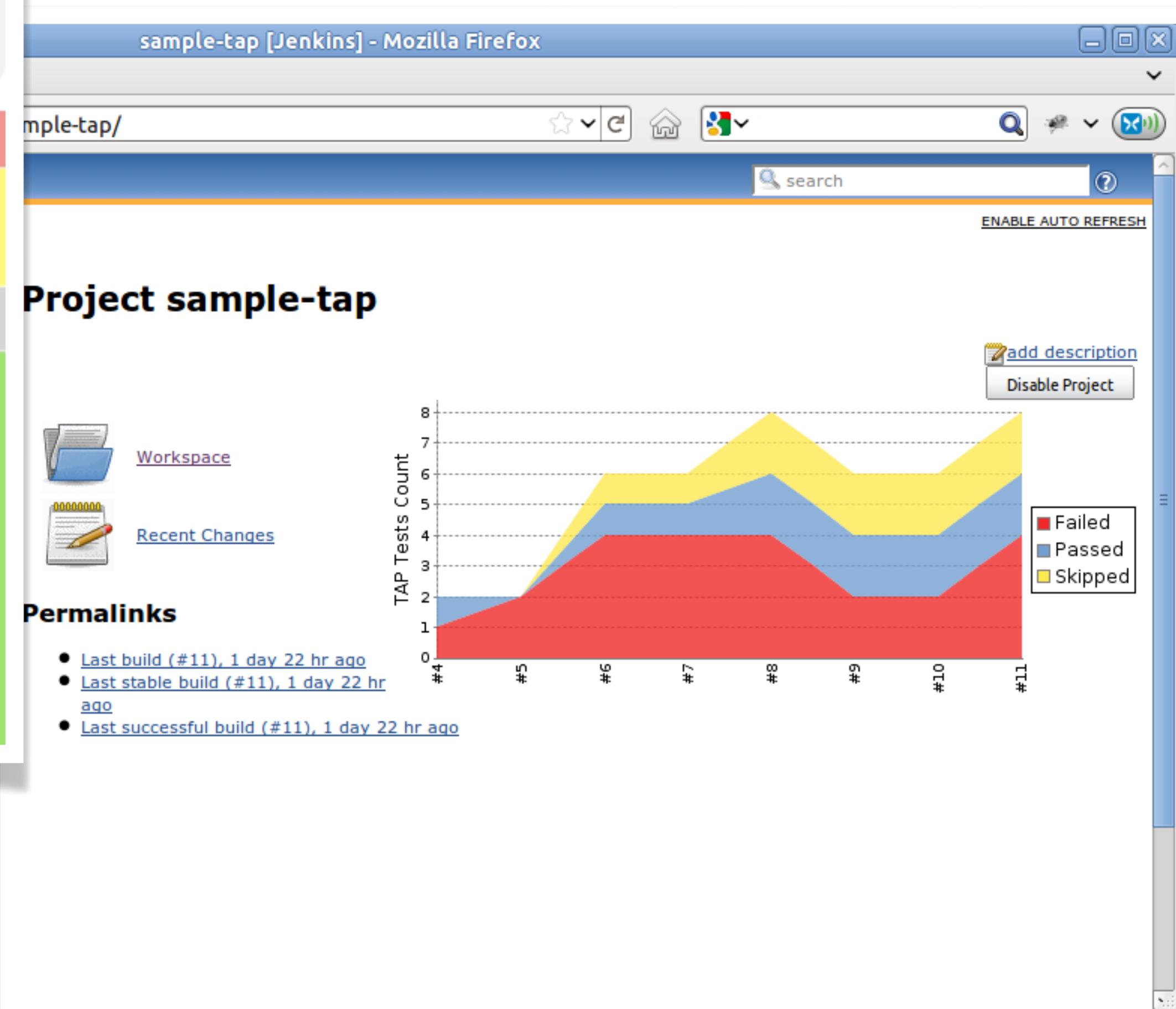
- 0 0 4 [Generic method return super type of type param and method parameter type param](#)
votes answers views [parameter type param](#)

StackOverflow, a knowledge base based on a reputation system

Additional tools with a social impact

- **Continuous integration** (with a strong testing culture) allows distributed development to happen with confidence that developers don't "break the build"
- **Code review** applications

Continuous integration



Code review on GitHub

 **larsmans** started a discussion in the diff 2 days ago

View full changes

doc/datasets/mldata_fixture.py

```
... ... @@ -10,7 +10,7 @@
10 10 import shutil
11 11
12 12 from sklearn import datasets
13 -from sklearn.utils.testing import mock_urllib2
13 +from sklearn.utils.testing import mock_mldata_urlopen
```

2

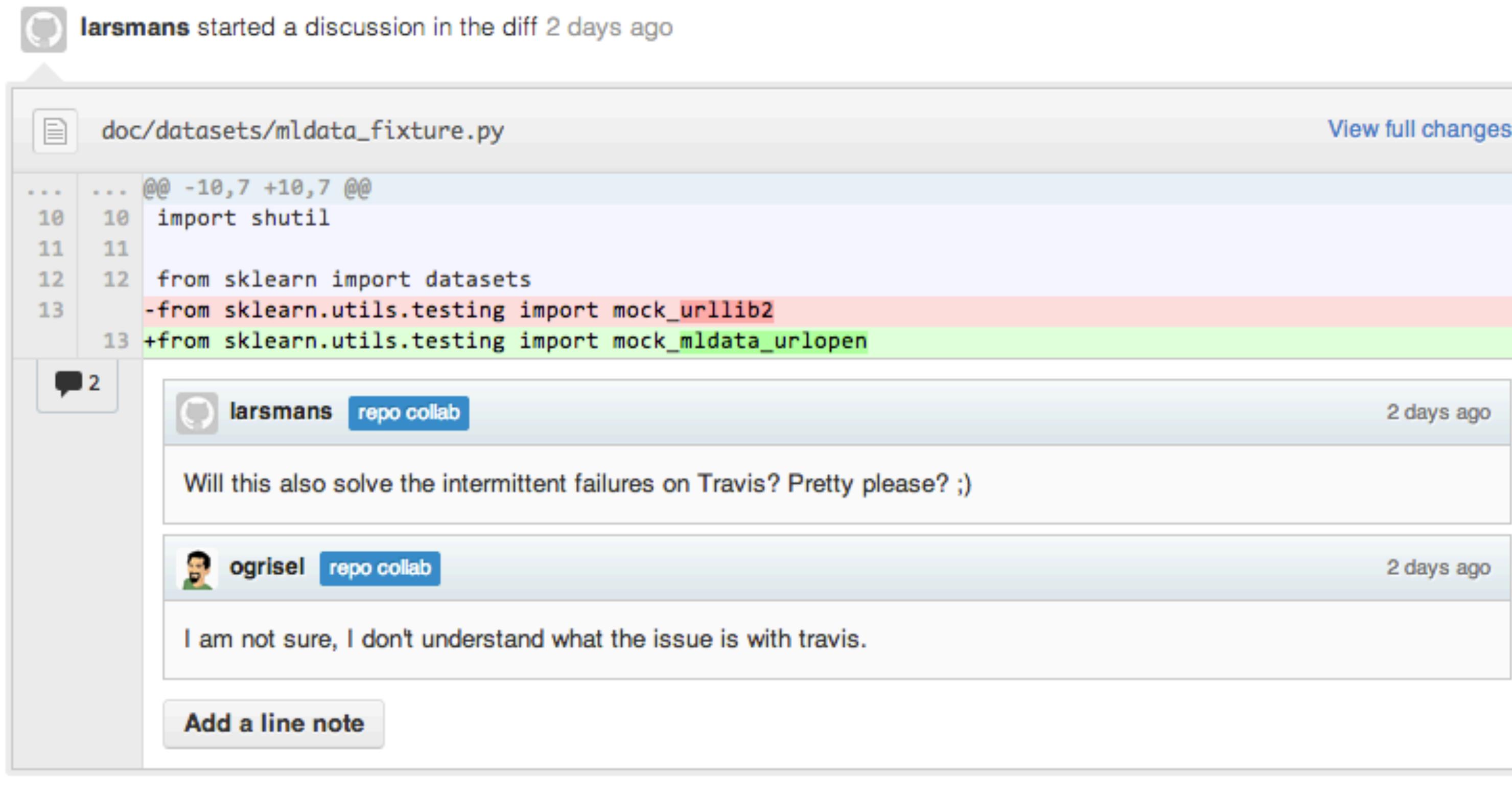
 **larsmans** repo collab 2 days ago

Will this also solve the intermittent failures on Travis? Pretty please? ;)

 **ogrisel** repo collab 2 days ago

I am not sure, I don't understand what the issue is with travis.

Add a line note



FOSS “2.0+(?)” (?)
2018-now

- GitHub est devenu quasiment incontournable, par application de la loi de Metcalfe
- GitHub, qui avait développé un aspect “plateforme” (i.e. permettant l’intégration de services tiers: CI/CD, audit de code, *SDLC management...*) a fini par intégrer certains de ces services, entrant ainsi en concurrence avec son écosystème
- Le rachat de GitHub par Microsoft fait par ailleurs tiquer de nombreux acteurs, et on voit l’émergence de nouvelles plateformes ou la popularité croissante de plateformes existantes

Modèles de maturité

Quantifier les pratiques - une bonne idée ?

Existing methodologies [edit]

There are more than 20 different OSS evaluation methods.^[1]

- **Open Source Maturity Model (OSMM)** from Capgemini
- **Open Source Maturity Model (OSMM)** from Navica^[2]
- Open Source Maturity Model (OSSMM) by Woods and Giuliani^[3]
- Methodology of Qualification and Selection of Open Source software (**QSOS**)
- **Open Business Readiness Rating (OpenBRR)**
- Open Business Quality Rating (OpenBQR)^[4]
- QualiPSO^[5]
- QualiPSO Model for Open Source Software Trustworthiness (MOSST)^{[6][7]}
- Towards A Trustworthiness Model For Open Source Software: How to evaluate Open Source Software^[8]
- QualOSS – Quality of Open Source^[9]
- Evaluation Framework for Open Source Software^[10]
- A Quality Model for OSS Selection^[11]
- Atos Origin Method for Qualification and Selection of Open Source Software (QSOS)^[12]
- Observatory for Innovation and Technological transfer on Open Source software (OITOS)^[13]
- Framework for OS Critical Systems Evaluation (FOCSE)^[14]

Comparison of open source maturity models

Umm-e-Laila, Adnan Zahoor, Khalid Mehboob , Sarfaraz Natha*

Sir Syed University of Engineering and Technology, CED , Gulshan -e- Iqbal, Karachi, Pakistan

Abstract

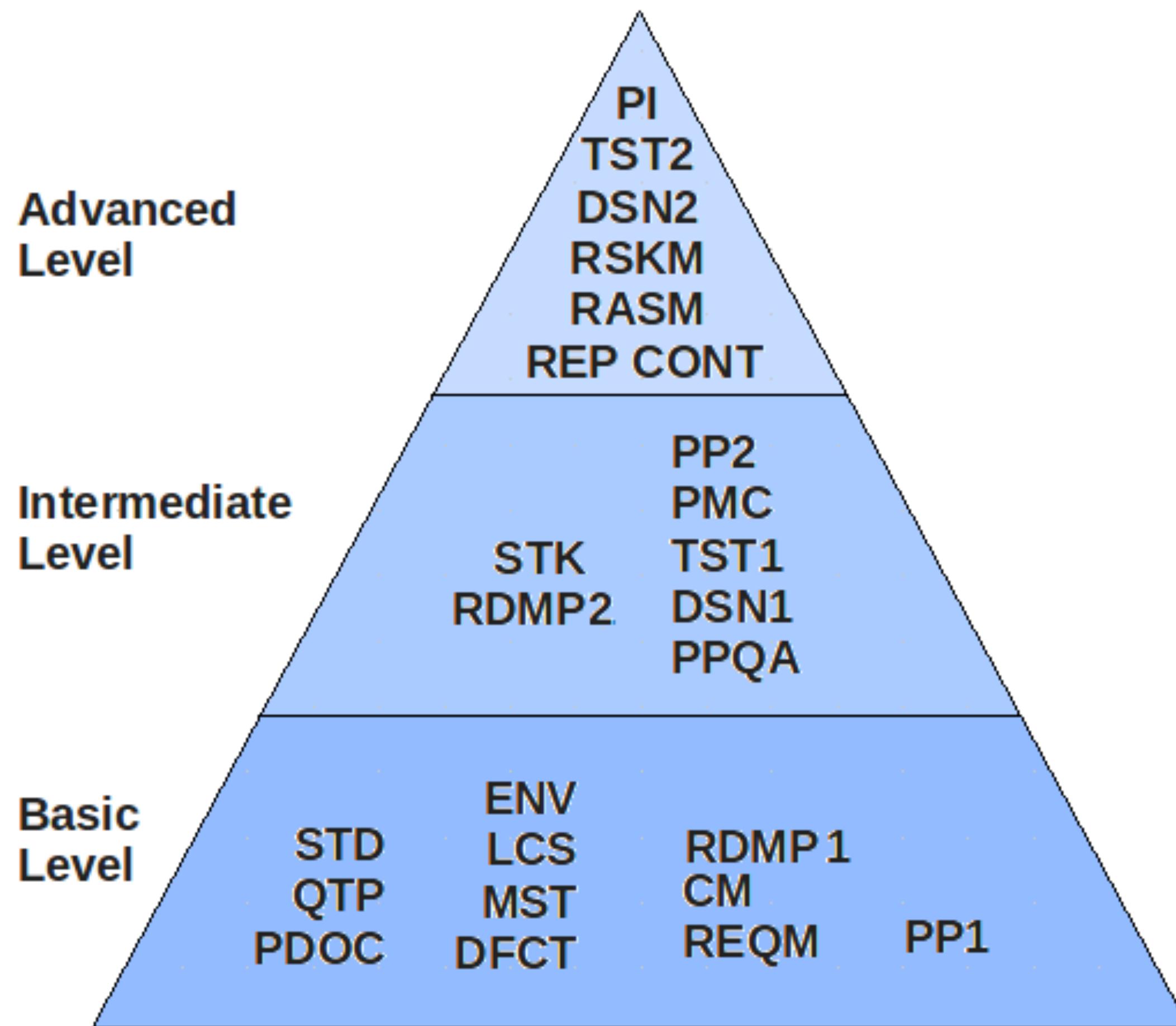
Component based development was formerly dependent on proprietary/closed source software's (CSS) components. Open Source software components has attracted noteworthy attention and become an operational alternative of proprietary software because of OSS security, cost effectiveness, quality, flexibility and freedom. Due to the increased attention on component-based development in the past decades, companies have widely adopted open source software (OSS), with the view that using the right software is critical to project success. The availability of Internet as a marketplace for components and wide adoption of OSS has introduced new challenges for selection of software components. Source Forge, other general and domain specific software repositories, different software foundations and individual OSS providers offer an abundance of OSS components. Identification, evaluation and selection of best possible OSS Components for the required need is a quite challenging job. As a reaction to these challenges different methods have been proposed for OSS maturity measurements. E.g. Capgemini-Open Source Maturity Model(C-OSSM), Navicasoft-Open Source Maturity Model (N-OSSM), Qualification and Selection of Open Source (QSOS), Open Business Readiness Rating (Open BRR) and Easiest Open Source (E-OSS).In this paper we compare different Open Source software maturity models available in the market that will help user in OSS component selection.

Modèles étudiés

Table 4. Comparison of oss maturity models

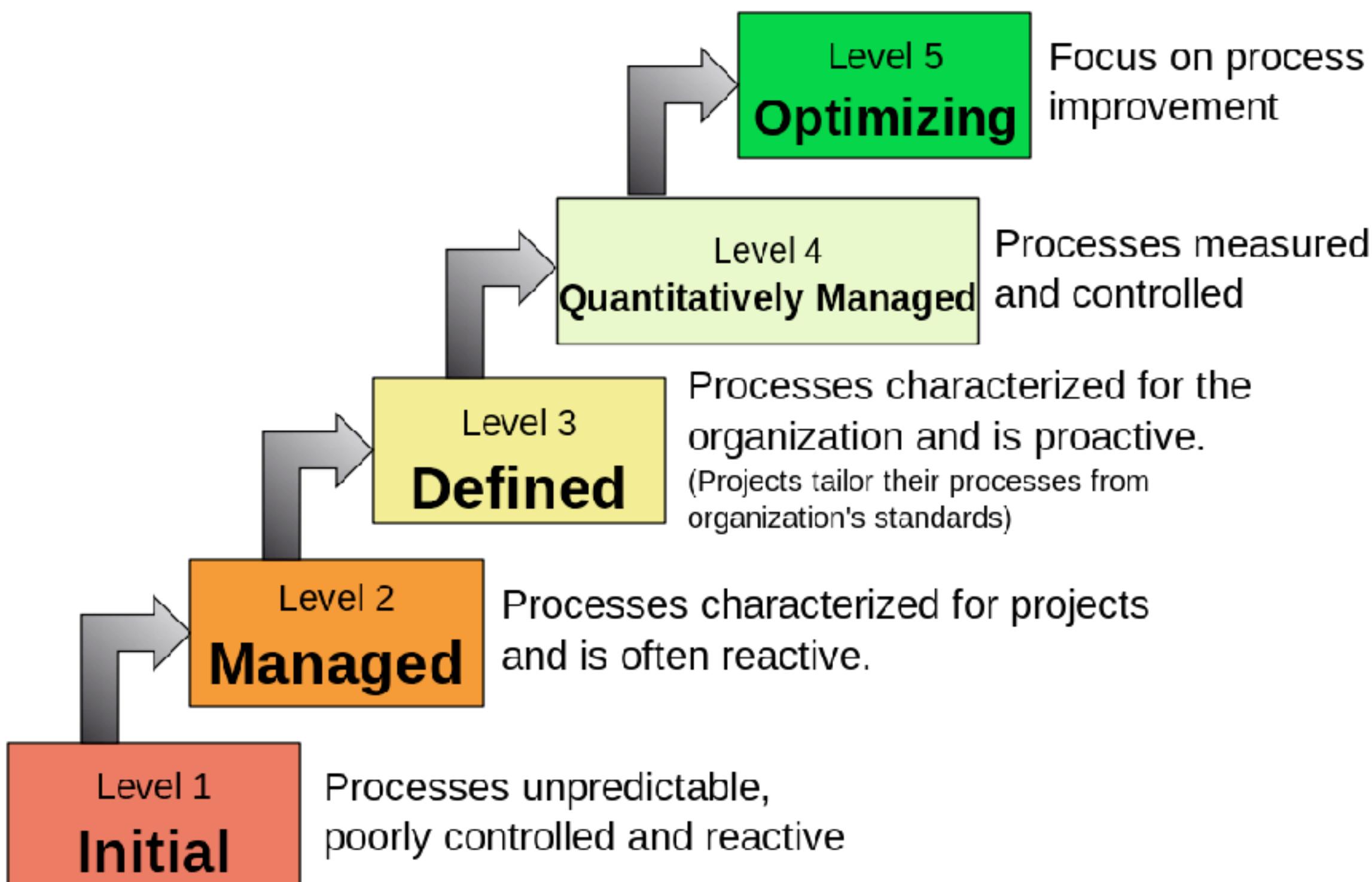
Acronym	Name	Corp/Org	Year	Strengths
C-OSMM	Capgemini Open Source Maturity Model	Cap Gemini	2003	Contain product and application indicators. Updation can be on regular basis via feedback from customers.
N-OSMM	Navica-Open Source Maturity Model	Navica Software	2004	Availability and Simplicity of tools for evaluation process.
QSOS	Methodology of Qualification and Selection of Open Source Software	Atos-Origin	2004	Consist of four iterative stages and supported by a tool called O3S
Open BRR	Open Business Readiness Rating	Carnegie Mellon West University, Spike Source, Intel and O'Reilly's	2005	Guarantee better decisions and increase confidence in selected open source software
E-OSS	Easiest Open Source Software Model	SIAD-Laboratory	2015	Easily adopt for small and medium business and also consider interoperability

OSMM (Capgemini / Qualipso)



(A comparer avec CMMI)

Characteristics of the Maturity levels



Capability Maturity Model Integration (CMMI) core process areas

Abbreviation	Process Area	Category	Maturity level
CM	Configuration Management	Support	2
MA	Measurement and Analysis	Support	2
PMC	Project Monitoring and Control	Project Management	2
PP	Project Planning	Project Management	2
PPQA	Process and Product Quality Assurance	Support	2
REQM	Requirements Management	Project Management	2
SAM	Supplier Agreement Management	Support	2
DAR	Decision Analysis and Resolution	Support	3
IPM	Integrated Project Management	Project Management	3
OPD	Organizational Process Definition	Process Management	3
OPF	Organizational Process Focus	Process Management	3
OT	Organizational Training	Process Management	3
RSKM	Risk Management	Project Management	3
OPP	Organizational Process Performance	Process Management	4
QPM	Quantitative Project Management	Project Management	4
CAR	Causal Analysis and Resolution	Support	5
OPM	Organizational Performance Management	Process Management	5

Basic level [edit]

The **basic level** that can be easily reached by adopting a few necessary practices in the FLOSS development process:

- **PDOC** – Product Documentation
- **STD** – Use of Established and Widespread Standards
- **QTP** – Quality of Test Plan
- **LCS** – Licenses
- **ENV** – Technical Environment
- **DFCT** – Number of Commits and Bug Reports
- **MST** – Maintainability and Stability
- **CM** – Configuration Management
- **PP1** – Project Planning Part 1
- **REQM** – Requirements Management
- **RDMP1** – Availability and Use of a (product) roadmap

Intermediate level [\[edit \]](#)

The **intermediate level** is the second level in OMM and can be achieved by fulfilling all trustworthy elements from the basic level and required trustworthy elements from the intermediate level.

- **RDMP2** – Availability and Use of a (product) roadmap
- **STK** – Relationship between Stakeholders
- **PP2** – Project Planning Part 2
- **PMC** – Project Monitoring and Control
- **TST1** – Test Part 1
- **DSN1** – Design Part 1
- **PPQA** – Process and Product Quality Assurance

Advanced level [edit]

The **advanced level** is the highest level that FLOSS projects can achieve by fulfilling all trustworthy elements from basic and intermediate levels and required trustworthy elements from the advanced level.

- **PI** – Product Integration
- **RSKM** – Risk Management
- **TST2** – Test Part 2
- **DSN2** – Design 2
- **RASM** – Results of third party assessment
- **REP** – Reputation
- **CONT** – Contribution to FLOSS Product from SW Companies

Modèle QSOS



How to Evaluate Open Source Software / Free Software (OSS/FS) Programs

David A. Wheeler

[dwheeler @ dwheeler.com](mailto:dwheeler@dwheeler.com)

Revised as of August 5, 2011

This paper describes a general process for evaluating programs, with specific information on how to evaluate Open Source Software / Free Software (OSS/FS) programs. This process is designed so that you can compare OSS/FS programs side-by-side with proprietary programs and other OSS/FS programs, and determine which one (if any) best meets your needs. This process is based on four steps: Identify candidates, Read existing reviews, Compare the leading programs' basic attributes to your needs, and then Analyze the top candidates in more depth. This set of Identify, Read Reviews, Compare, and Analyze can be abbreviated as "IRCA". Important attributes to consider include functionality, cost, market share, support, maintenance, reliability, performance, scaleability, useability, security, flexibility/customizability, interoperability, and legal/license issues.

Modèles plus récents

- Gouvernance → Good Governance Initiative (OW2 → OSPO Alliance)
- Sécurité → OpenSSF Best Practices Badge Program
- Souveraineté → European Sovereignty Software Index

OpenSSF Best Practices Badge Program

Get Your Badge Now!

The [Open Source Security Foundation \(OpenSSF\)](#) Best Practices badge is a way for Free/Libre and Open Source Software (FLOSS) projects to show that they follow best practices. Projects can voluntarily self-certify, at no cost, by using this web application to explain how they follow each best practice. The OpenSSF Best Practices Badge is inspired by the many badges available to projects on GitHub. Consumers of the badge can quickly assess which FLOSS projects are following best practices and as a result are more likely to produce higher-quality secure software.

You can easily see the [criteria for the passing badge](#). More information on the OpenSSF Best Practices Badging program is [available on GitHub](#). [Project statistics](#) and [criteria statistics](#) are available. The [projects page](#) shows participating projects and supports queries (e.g., you can see [projects that have a passing badge](#)). You can also see [an example](#) (where we try to earn our own badge). This project was formerly known as the Core Infrastructure Initiative (CII) Best Practices badge, and was originally developed under the CII. It is now part of the [OpenSSF Best Practices Working Group \(WG\)](#). The OpenSSF is a foundation of the [Linux Foundation \(LF\)](#). The project was formally renamed from "CII Best Practices badge" on 2021-12-24.

Privacy and legal issues: Please see our [privacy policy](#), [about cookies](#), and [terms of use](#). The code for the badging application itself is released under the [MIT](#) license (projects pursuing a badge are under their respective licenses). All publicly-available non-code content managed by the badging application is released under at least the [Creative Commons Attribution License version 3.0 \(CC-BY-3.0\)](#); newer non-code content is released under CC-BY version 3.0 or later (CC-BY-3.0+). If referencing collectively or not otherwise noted, please credit the OpenSSF Best Practices badge contributors.



Some badge earners:



CII Best Practices Badge



- Identifies best practices for OSS projects
 - Goal: Increase likelihood of better quality & security. E.g.:
 - “The project sites... MUST support HTTPS using TLS.”
 - “The project MUST use at least one automated test suite...”
 - “At least one static code analysis tool MUST be applied...”
 - “The project MUST publish the process for reporting vulnerabilities on the project site.”
 - Based on practices of well-run OSS projects
- If OSS project meets best practice criteria, it earns a badge
 - Enables projects & potential users know current status & where it can improve
 - Combination of self-certification, automated checks, spot checks, public accountability
 - Three badge levels: passing, silver, gold
- Participation widespread & continuing to grow
 - >3,700 participating projects, > 500 passing+ projects in 2021-04
 - Current statistics: https://bestpractices.coreinfrastructure.org/en/project_stats
- A project within the OpenSSF Best Practices Working Group (WG)
- For more, see: <https://bestpractices.coreinfrastructure.org>



abilian-sbe

[Expand panels](#) [Show all details](#) [Hide met & N/A](#)

Projects that follow the best practices below can voluntarily self-certify and show that they've achieved an Open Source Security Foundation (OpenSSF) best practices badge. [Show details](#)

If this is your project, please show your badge status on your project page! The badge status looks like this:

`openssf best practices in progress 60%` Here is how to embed it: [Show details](#)

These are the [passing](#) level criteria. You can also view the [silver](#) or [gold](#) level criteria.

▲ Basics 8/13 •

Identification

What is the human-readable name of the project? [Show details](#)

abilian-sbe

What is a brief description of the project?

Abilian Social Business Engine - an enterprise social networking / collaboration platform.

What is the URL for the project (as a whole)?

<http://www.abilian.com>

What is the URL for the version control repository (it may be the same as the project URL)?

<https://github.com/abilian/abilian-sbe>

What programming language(s) are used to implement the project? [Show details](#)

Python

What is the Common Platform Enumeration (CPE) name for the project (if it has one)? [Show details](#)

(Optional) CPE name

Basic project website content



- Met
- Unmet
- ?

The project website **MUST** succinctly describe what the software does (what problem does it solve?). [description_good] [Show details](#)



- Met
- Unmet
- ?

The project website **MUST** provide information on how to: obtain, provide feedback (as bug reports or enhancements), and contribute to the software. [interact]



- Met
- Unmet
- ?

The information on how to contribute **MUST** explain the contribution process (e.g., are pull requests used?) (URL required) [contribution] [Show details](#)

Projects on GitHub by default use issues and pull requests, as encouraged by documentation such as <https://guides.github.com/activities/contributing-to-open-source/>.



- Met
- Unmet
- ?

The information on how to contribute **SHOULD** include the requirements for acceptable contributions (e.g., a reference to any required coding standard). (URL required) [contribution_requirements]

FLOSS license

What license(s) is the project released under? [Show details](#)

LGPL-2.1

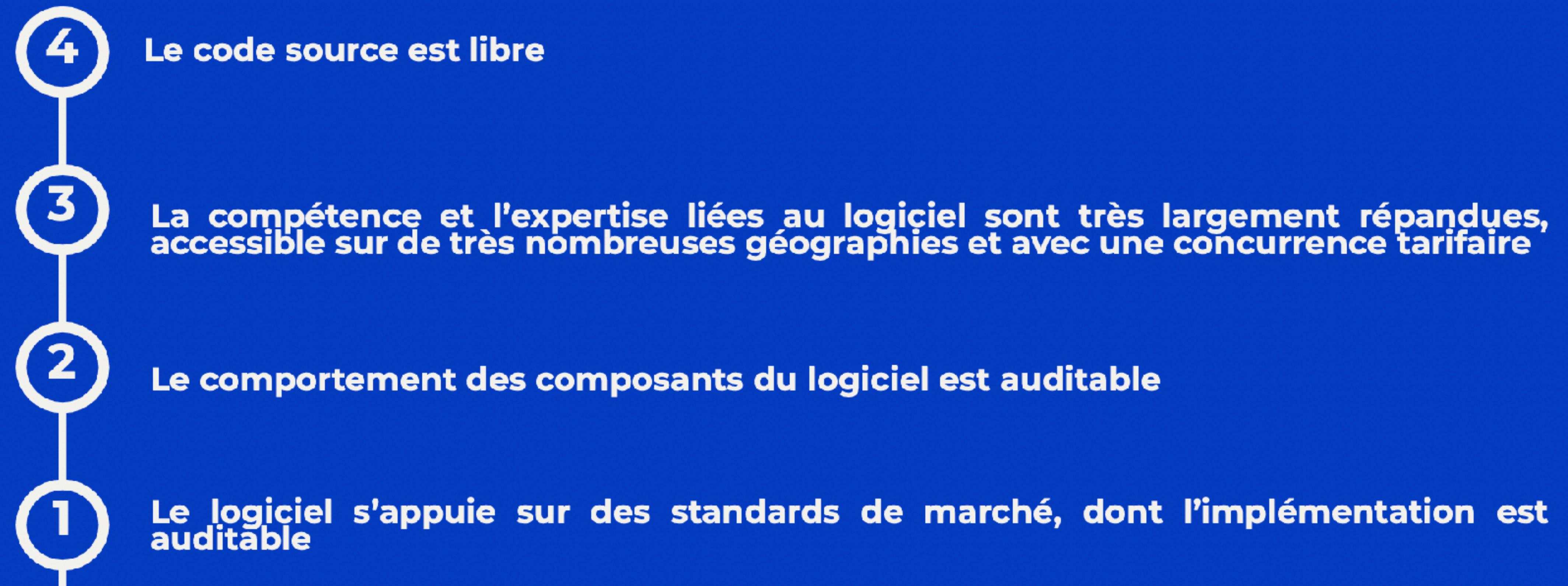
4 axes d'analyse pour estimer la contribution du choix d'un logiciel sur la souveraineté numérique européenne

Source: Présentation de Denis Lafont-Trevisan à l'OSXP 2022

Axe 1 - Maîtrise Juridique

- 
- 1 Le logiciel permet un hébergement des données dans des environnements juridiques en nombre restreint.
 - 2 Le logiciel permet un hébergement des données et des traitements dans des environnements juridiques en nombre restreint.
 - 3 La propriété intellectuelle du logiciel est détenue par une entité légale dont la législation limite les atteintes à la souveraineté. Le logiciel permet une execution dans un environnement dont on peut choisir le cadre légal sans contrainte ou risque juridique fort.
 - 4 La propriété intellectuelle du logiciel est partagée entre plusieurs acteurs indépendants et/ou permet de limiter les risques d'actions juridiques hostiles. Le code peut être exécuté (données+calculs+mémoire) sur un environnement dont on peut choisir le cadre légal. Le logiciel est immune au droit extra-européen.

Axe 2 - Maîtrise technologique

- 
- 4 Le code source est libre
 - 3 La compétence et l'expertise liées au logiciel sont très largement répandues, accessible sur de très nombreuses géographies et avec une concurrence tarifaire
 - 2 Le comportement des composants du logiciel est auditabile
 - 1 Le logiciel s'appuie sur des standards de marché, dont l'implémentation est auditabile

Axe 3 - Réversibilité

- 
- 1 La chaîne de dépendances est complexe
 - 2 La chaîne de dépendances est connue et maîtrisable. Les données sont portables. Les documentations existent et sont exploitables en conditions nominales
 - 3 Les règles métiers sont documentées et/ou exportables. La documentation et l'historique des versions avec le changelog sont accessibles
 - 4 Il existe une grande communauté d'utilisateurs et consultants actifs et mobilisables pour le support, et en coopération

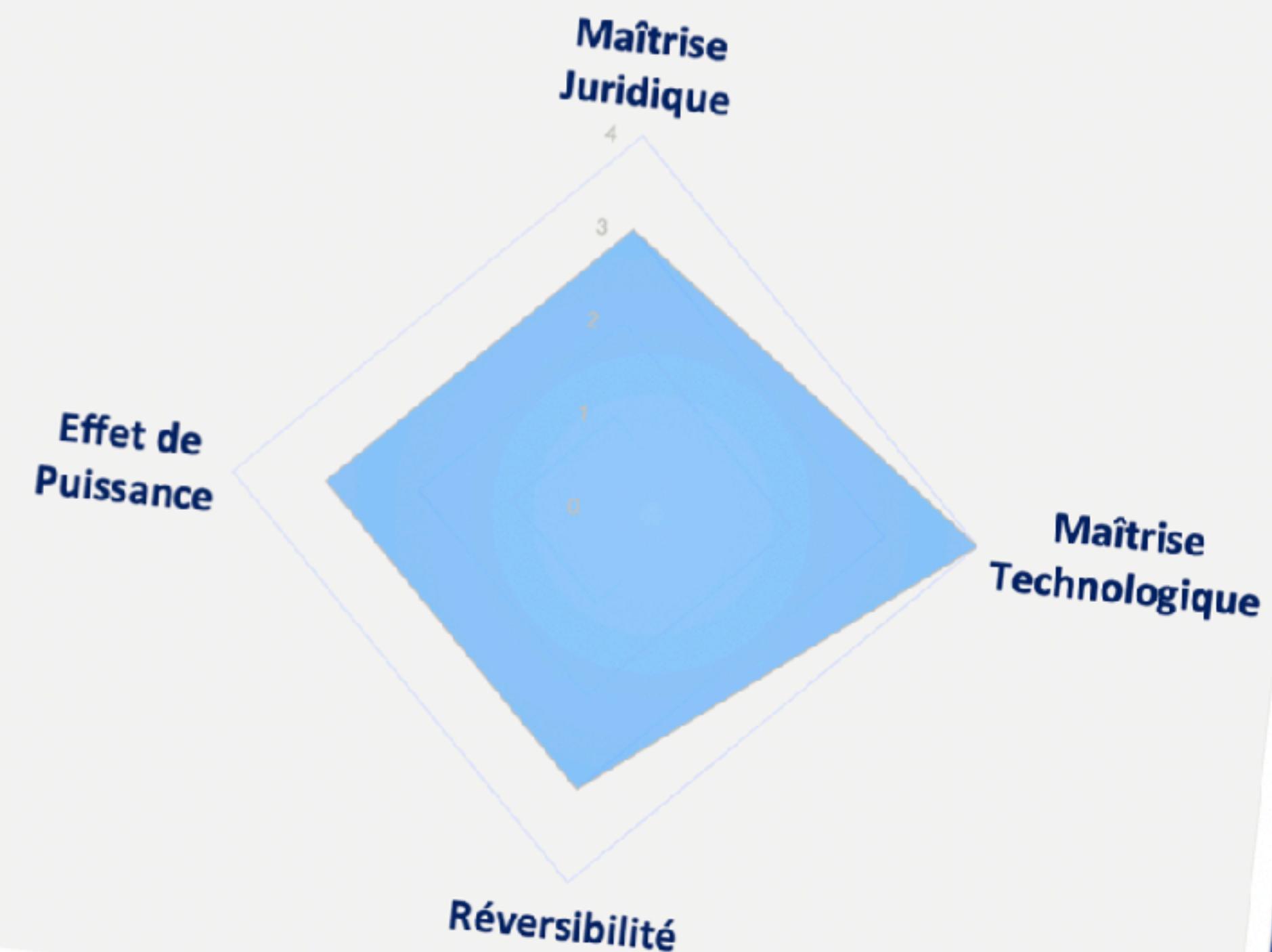
Axe 4 - Effet de puissance

- 4 L'écosystème autour du logiciel est constitué de très nombreux acteurs européens
- 3 Le logiciel est détenu par des entités (Fondateurs, VC, entreprises, Consortium...) essentiellement européens
- 2 Les créateurs du logiciel sont majoritairement européens
- 1 Les créateurs du logiciel sont en partie européens

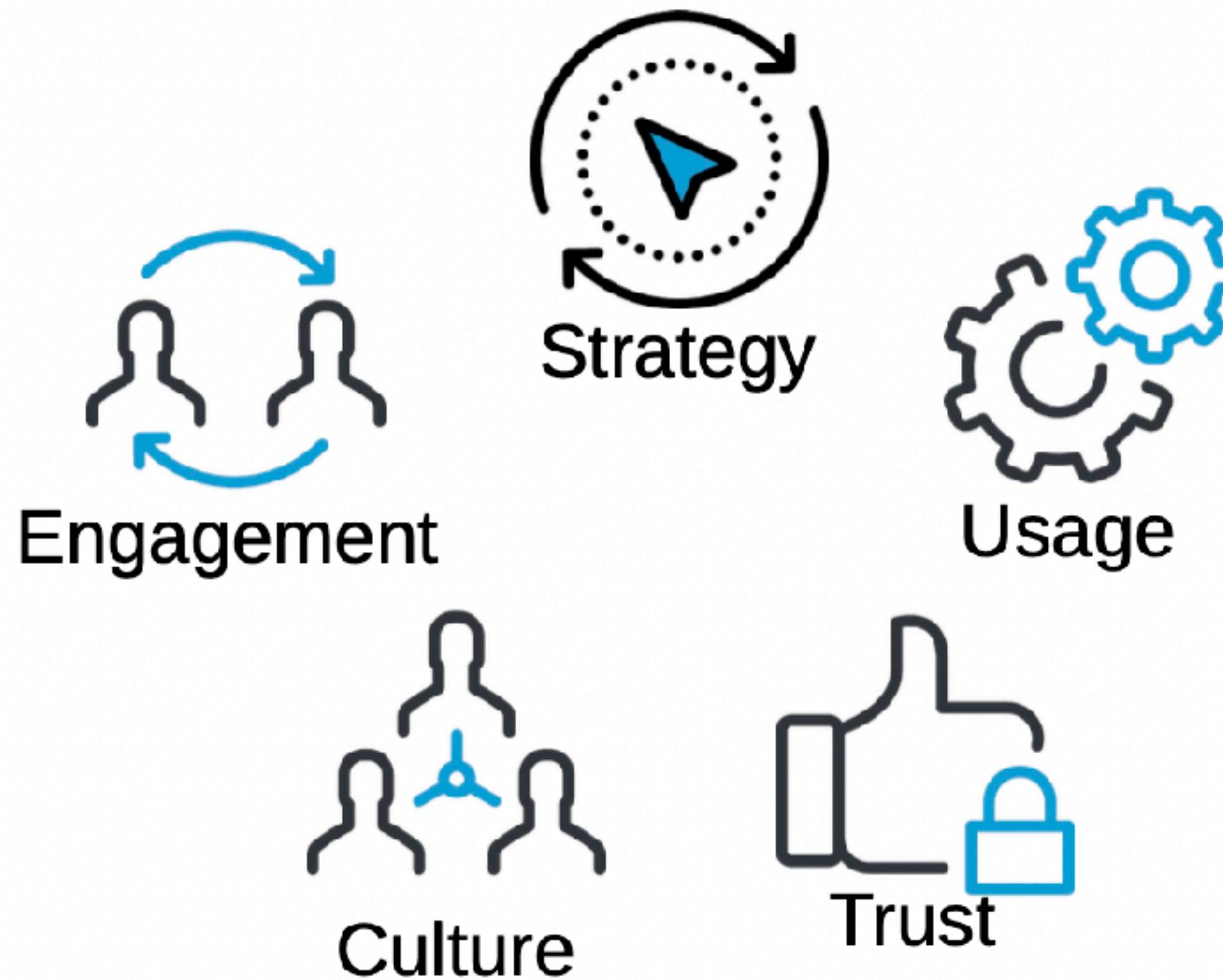
NGINX



 **traefik**
proxy



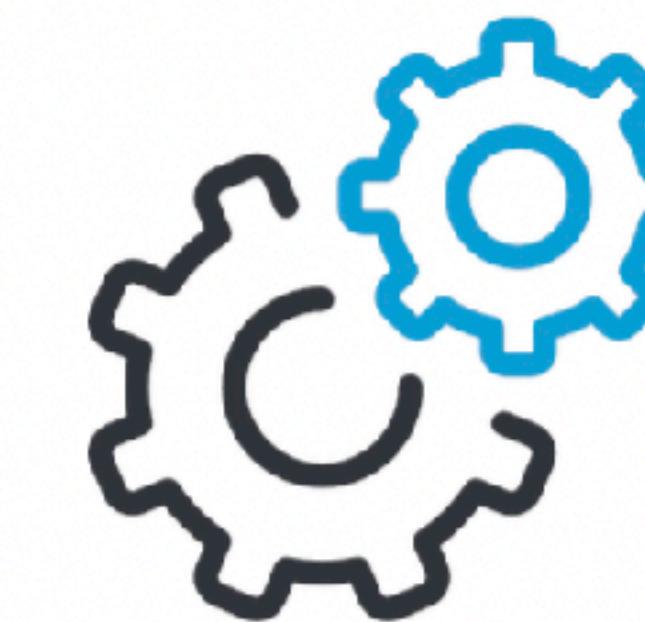
A Comprehensive Framework for Managing Open Source Software



- ▶ A heuristic approach to open source
- ▶ A blueprint for open source policy implementation
- ▶ Five complementary **GOALS**
- ▶ Supported by **CANONICAL ACTIVITIES**
- ▶ Verified by **CUSTOMIZED SCORECARDS**

Usage Goal

Technical Skills in Using Open Source Software.



► Key content

- Using OSS and competences in using OSS. It covers technical ability and experience with OSS
- Ensuring some understanding and awareness of OSS

► Current activities:

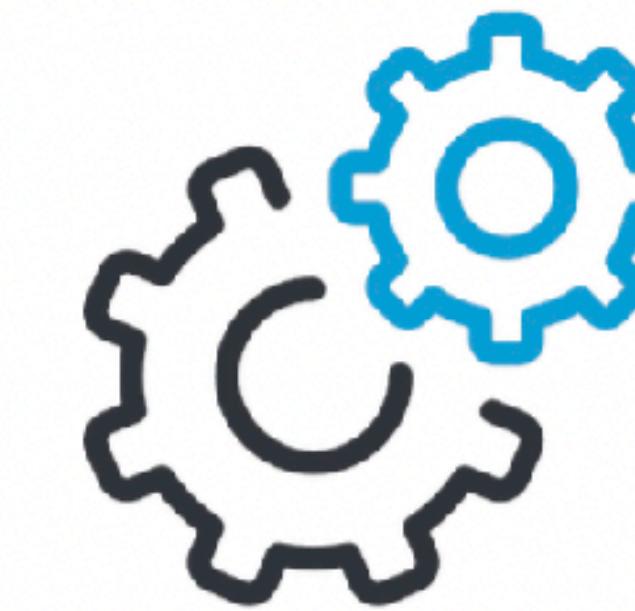
- Managing OSS development skills
- Ensure basic skills in OSS technos
- Proactive OSS decision making
- Using OSS enterprise software
- OSS review

► Verification points

- Inventory of OSS used
- Inventory of available skills
- Inventory of required skills
- Understanding of OSS licenses

Usage Goal

Technical Skills in Using Open Source Software.



► Key content

- Using OSS and competences in using OSS. It covers technical ability and experience with OSS
- Ensuring some understanding and awareness of OSS

► Current activities:

- Managing OSS development skills
- Ensure basic skills in OSS technos
- Proactive OSS decision making
- Using OSS enterprise software
- OSS review

► Verification points

- Inventory of OSS used
- Inventory of available skills
- Inventory of required skills
- Understanding of OSS licenses

Trust Goal

Secure and Appropriate Usage of Open Source Software.



► Key content

- Using OSS legally and safely
- Organised software reuse
- Organised license management
- OSS production best practices

► Current activities:

- Managing legal compliance
- Manage software vulnerabilities
- Manage software dependencies
- Manage key indicators

► Verification points

- Compliance policy, officer
- Dependency and security management policy
- Use of compliance and dependency management tools

Culture Goal

Belonging to the Open Source Community at Large.



► Key content

- OSS culture that helps implementing best practices
- Sharing with the open source community
- Engaging with support communities

► Current activities:

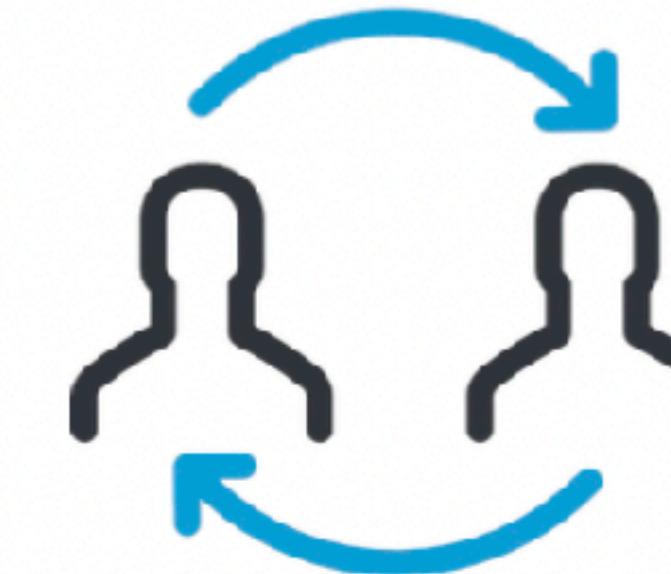
- OSS development principles
- Contributing to OSS projects
- Upstream first
- HR perspective
- Belonging to the OSS community

► Verification points

- Personal and Professional contributions to OSS projects
- Individual attendance at OSS events

Engagement Goal

Engaging with the OSS ecosystem.



► Key content

- ▶ Contributing back to OSS projects and supporting OSS communities
- ▶ Publicly asserting use of open source

► Current activities:

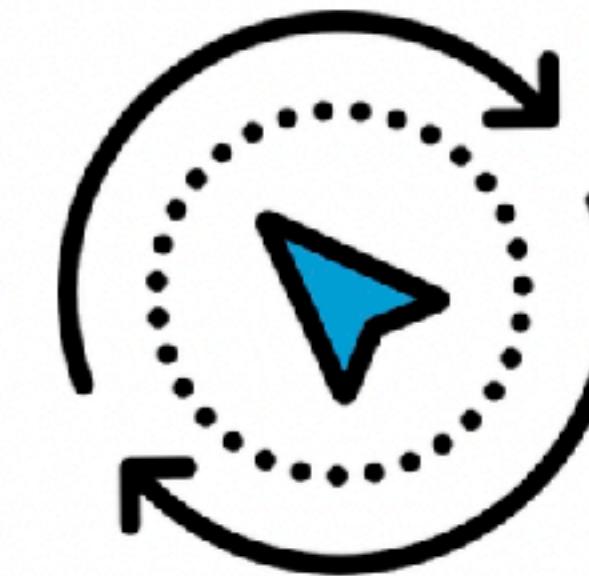
- ▶ Contributing back to OSS projects
- ▶ Supporting OSS communities
- ▶ Publicly asserting use of OSS
- ▶ Contracting with OSS vendors
- ▶ Participating in OSS events

► Verification points

- ▶ Communication actions
- ▶ Event participation
- ▶ Financial sponsoring
- ▶ Publicly shared code and contributions

Strategy Goal

Proactively using OSS for innovation and competitiveness.



►Key Content

- ▶ C-level awareness of open source as an enabler
- ▶ Innovation
- ▶ Sovereignty
- ▶ Digital transformation

►Current activities:

- ▶ C-Level awareness building
- ▶ Link OSS with digital sovereignty
- ▶ OSS enabling innovations
- ▶ OSS enabling digital transformation

►Verification points

- ▶ Open source in corporate IT strategy
- ▶ Open source supporting new activities
- ▶ Open source in annual report

Conseils pratiques et activités dans le contexte d'un éditeur open source

People first

- Give warm welcomes to new members
- Thank contributors
- Give positive feedback
- Act quickly on new contributions (thank you, feedback, commit)
- Never forget to give credit (CONTRIBUTORS.txt, release notes)

Make it easy to become a contributor

- It should be easy to add or fix a translation, a particular bit of documentation, a FAQ entry, etc.
- It should also be easy to contribute new modules (add-ons)
- This is the whole idea of “**The architecture of participation**” (O'Reilly, 2004)

When to give away the “commit bit”

- New contributors have to go through a learning process and build trust before being allowed to commit directly on the code repository
- Ask them first to submit patches on the issue tracker
- Some legal paperwork can be required (CLA / CCA / DCO)
- Practices vary widely between different projects

Engage with people

- Be generic:
 - Solicit feedback (“what do you think of...?”)
 - Ask for beta testers, bug reports
- Be specific:
 - Link to the right places (relevant space on issue tracker, forum, FAQ entry, etc.)
 - Engage with specific people

The Roadmap

- Make the roadmap clear and visible
 - Publish plan for at least next minor and major releases
 - Include *tentative* dates and scope (make it clear it *is* tentative, though)
- Make it consistent with the Issue Tracker (and the reality)
- Ask for feedback and contributions

Get good at Email

- (Assuming you're using a mailing list; this is probably similar w/ a forum like Discourse)
- Reformulate until everything's 100% clear
- Make your emails easy to read (short paragraphs, skip one line btw paragraphs...)
- Don't over quote previous messages, but keep some context
- Use URLs to quote previous conversations or online documents

Blog

- Some email messages (new features, etc.) should be written as blog posts, *then* sent to the mailing list (either copied or as links)
- Put pictures or diagrams on your blog posts
- Weekly / monthly technical reports
- Reinforce with tweets and other status updates

FAQ and READMEs

- There should be one README in each project module (even if it's only one link to a particular web page)
- Read “README-driven development” (by Tom Preston-Werner):
 - *“Write your Readme first. First. As in, before you write any code or tests or behaviors or stories or ANYTHING. I know, I know, we’re programmers, dammit, not tech writers! But that’s where you’re wrong. Writing a Readme is absolutely essential to writing good software.”*
 - Constantly update the FAQ with questions asked on the mailing list or feedback from the community

Community vs. Support

- If someone's obviously using the community as a substitute for support, let others deal with him
- Don't support people that never give anything in return
- Aggressive people should be dealt with with care, and certainly not by being aggressive in return

Community vs. Sales

- When you identify interesting people in the community, pass useful information to sales
- Sometimes hint that we are doing interesting projects for real customers (without giving away confidential information)
- Give information to help people make their case for using the product in their organization

Make it easy to build our software

- Indispensable pour avoir des contributeurs (et des utilisateurs)
- Old way:
 - “./configure ; make ; make install”
- New way:
 - “pip install .”, “mvn install”, “npm install”, “go build”, etc.
 - Or docker / docker-compose

Provide good documentation

- C'est un métier ("Technical writer"), mais aussi une activité à ne pas négliger pour les développeurs
- Intégration avec le code ("API doc", etc.)
- Outils populaires: Sphinx, Mkdocs, etc.
- Sources:
 - <https://www.writethedocs.org/>
 - Books

Practice open decision making

- Processus de décisions transparents, concernant notamment:
 - Les corrections et améliorations mineures (bug trackers publics)
 - Les améliorations substantielles (ex: “PEPs = Python Enhancement Proposals”)
 - La nomination des personnes décisionnaires

Dealing with poisonous people

- “Code of conduct” / “Charters”
- Enforcement
- Cf. Cathy Sierra évoqué en début de présentation

Relationship with other communities

- Upstream: languages, libraries
- Sidestream: plugins creators
- Downstream: packagers (e.g. for Linux distributions)
- Customers: OSPOs

Recap

- It's about **people, first**: getting to know each other, making sense of the crowd, creating a sense of belonging
- Always be **respectful, transparent, authentic** and **helpful**
- Contribute to the **architecture of participation**

References / Credits

- <https://producingoss.com/> (Karl Fogel)
- <http://www.artofcommunityonline.org/> (Jono Bacon)
- <http://www.slideshare.net/bdelacretaz/life-in-open-source-communities-apachecon-us-2009> (Bertrand Delacretaz)
- <http://www.slideshare.net/jaaronfarr/making-open-source-work-presentation> (J. Aarron Farr)
- <http://lwn.net/Articles/370157/> (Josh Berkus)
- <http://headrush.typepad.com/> (Kathy Sierra)