Supporting and Assessing Market Readiness of OW2 Projects

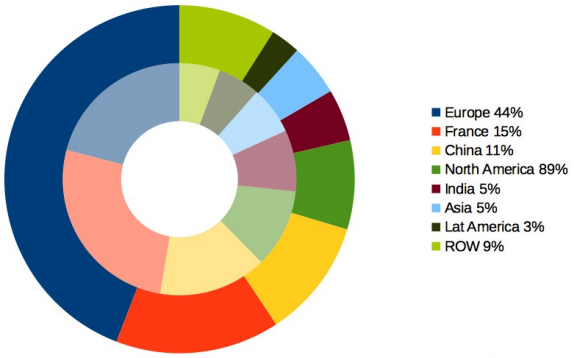
A Progress Report

Cédric Thomas, OW2

FOSDEM, Saturday, February 4, 2017

OW2
Non-Profit Open Source Organization
European and Global

Community

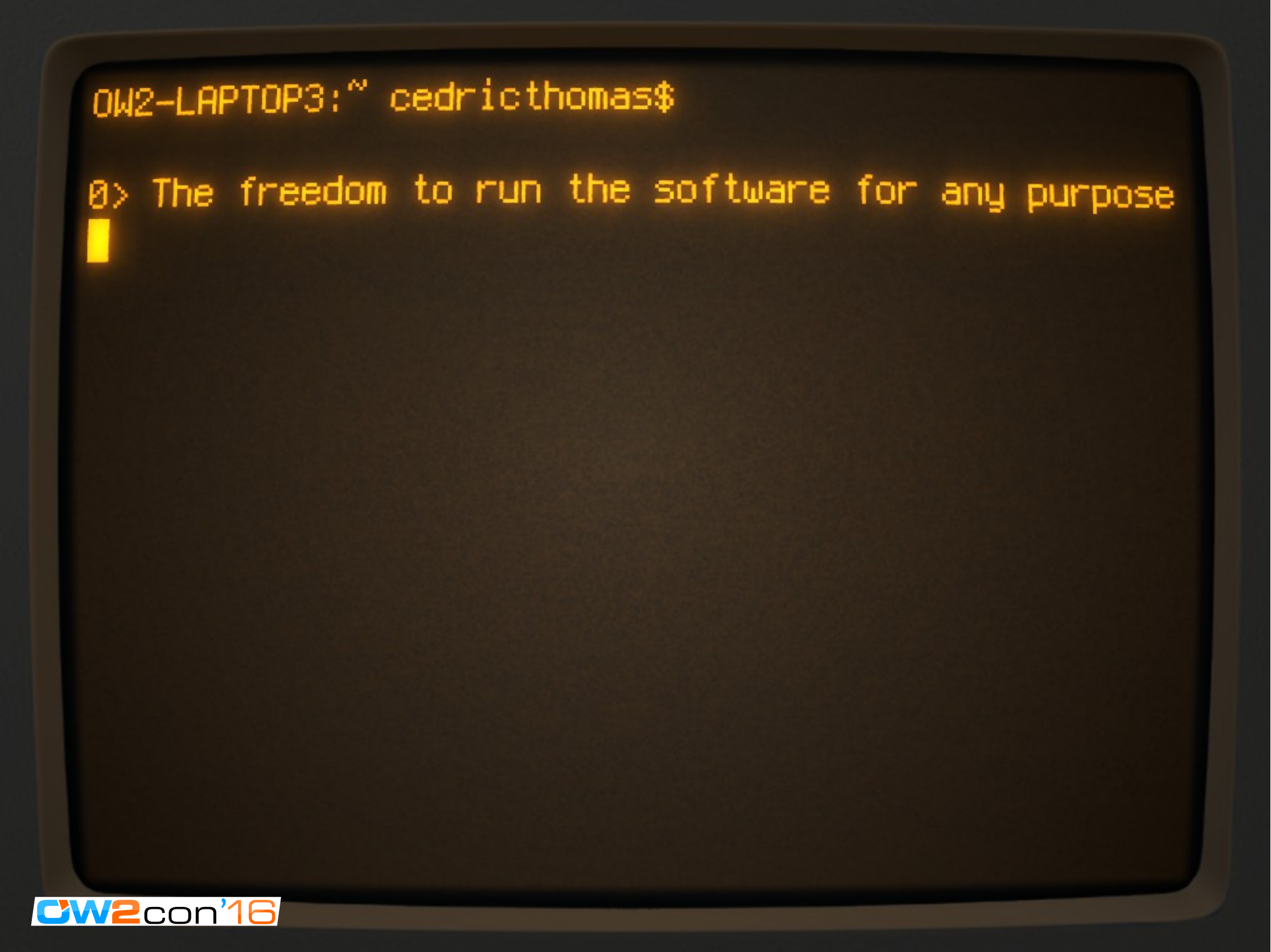


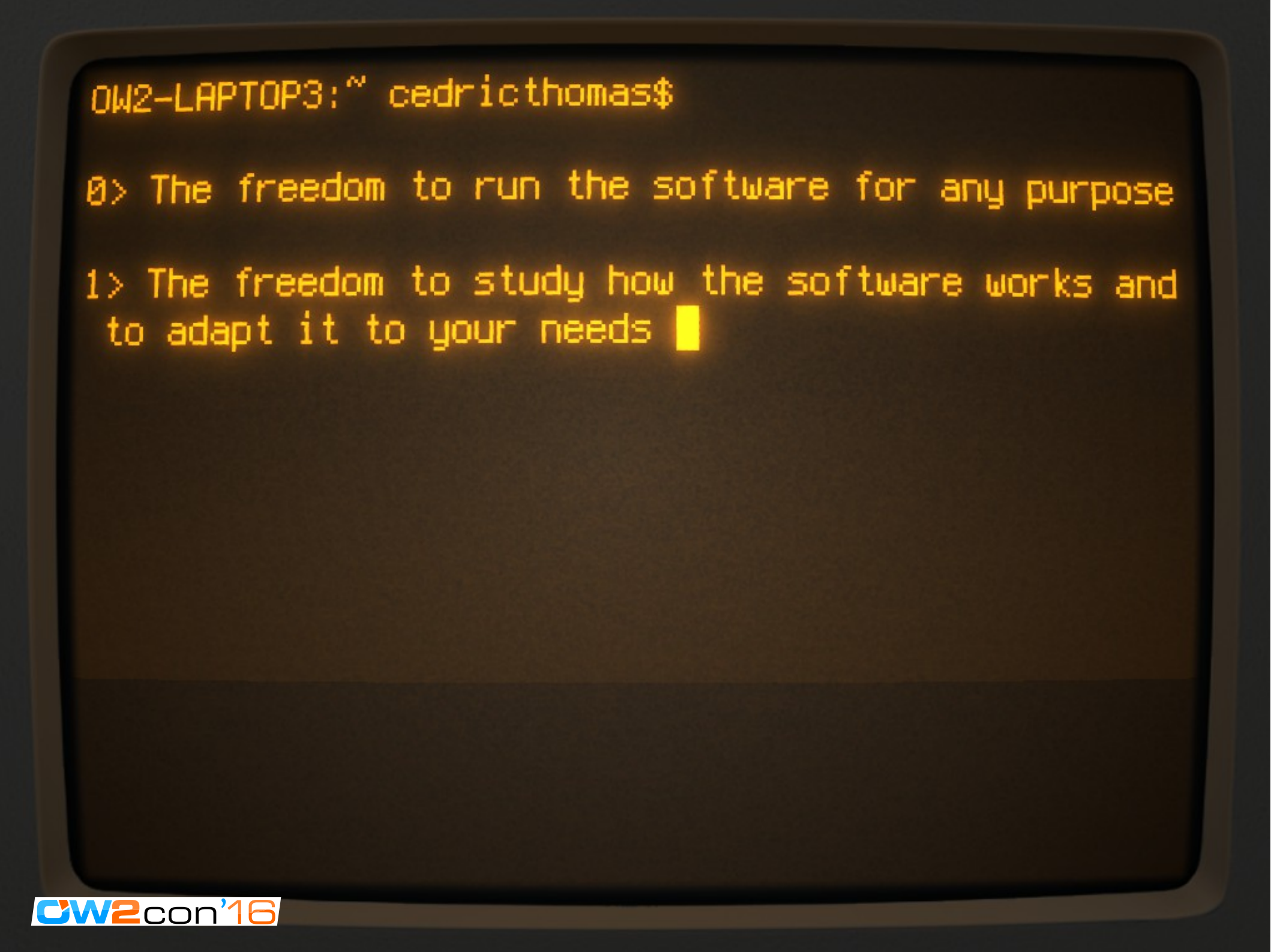
Activities

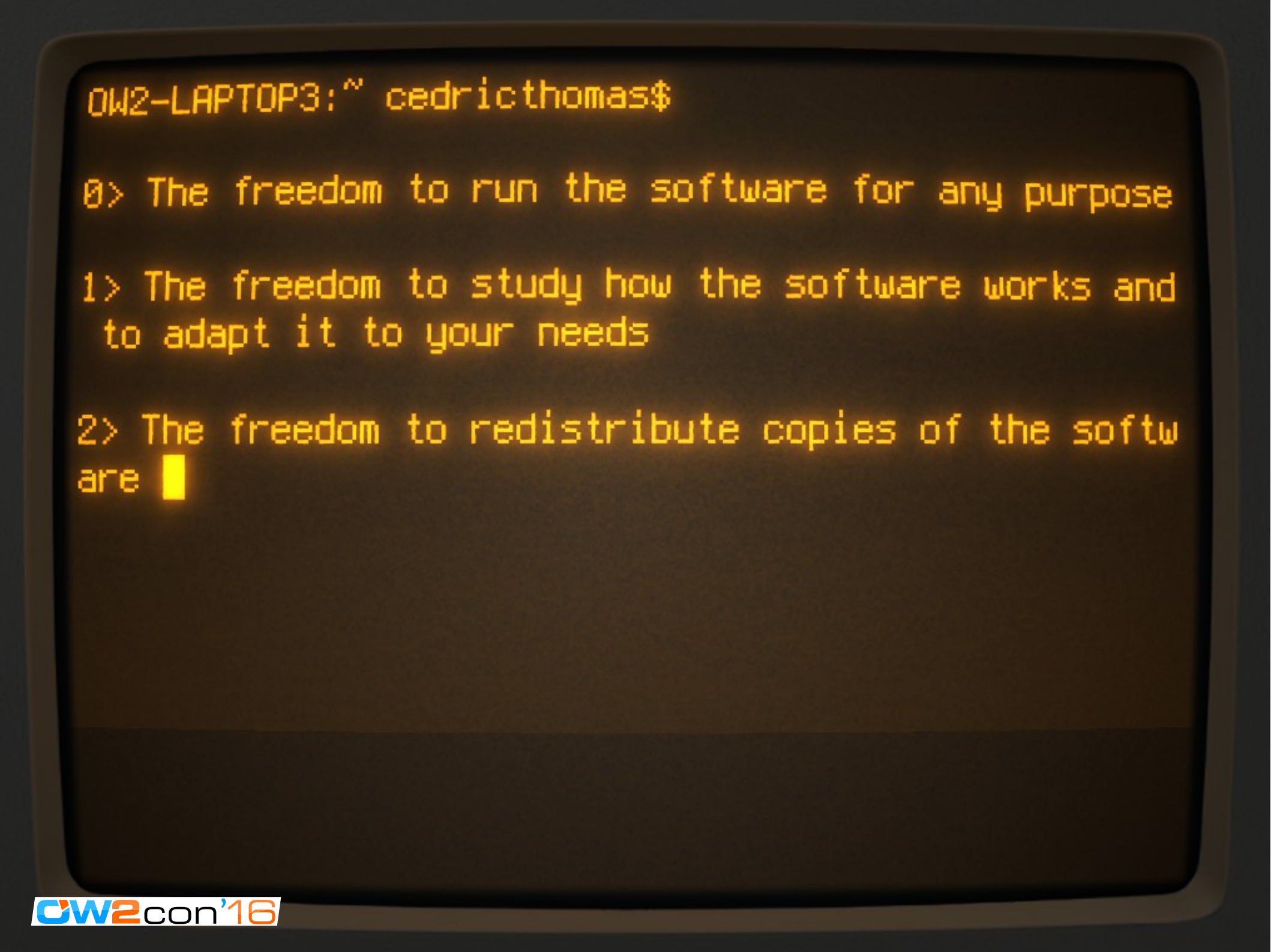
Members

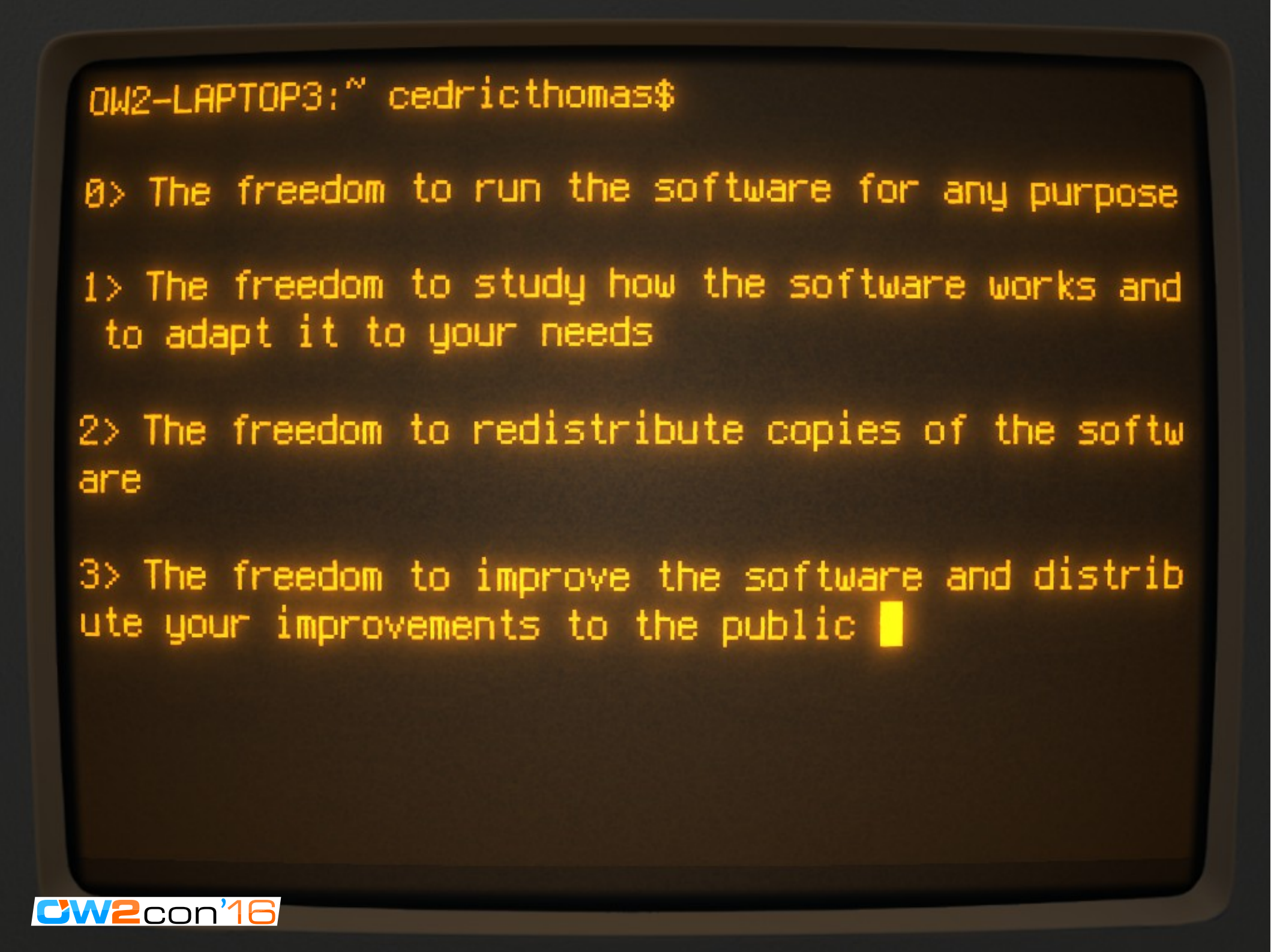
Code Base

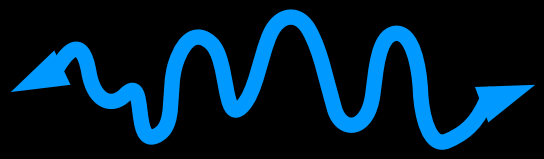
Governance


OW2-LAPTOP3:~ cedricthomas$
0> The freedom to run the software for any purpose

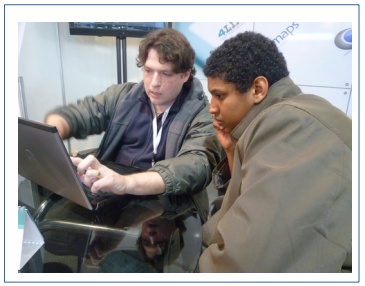

OW2-LAPTOP3:~ cedricthomas$
0> The freedom to run the software for any purpose
1> The freedom to study how the software works and to adapt it to your needs

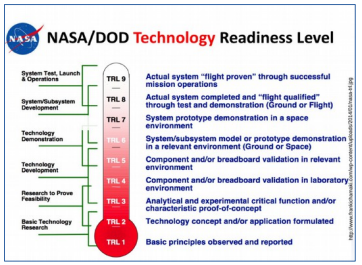

OW2-LAPTOP3:~ cedricthomas$
0> The freedom to run the software for any purpose
1> The freedom to study how the software works and to adapt it to your needs
2> The freedom to redistribute copies of the softw are

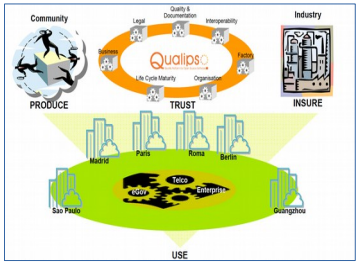

OW2-LAPTOP3:~ cedricthomas$
0> The freedom to run the software for any purpose
1> The freedom to study how the software works and to adapt it to your needs
2> The freedom to redistribute copies of the softw are
3> The freedom to improve the software and distrib ute your improvements to the public

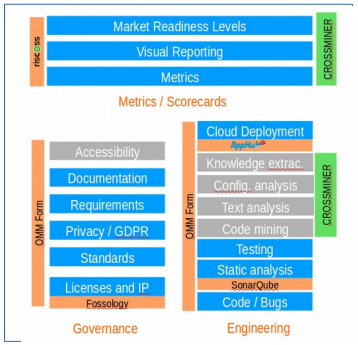
Free Software  Commercial Open Source

Agenda









- OSS Projects and the Value Chain
- Evaluating Readiness and Maturity
- Evaluating Open Source Projects
- OW2 OSCAR Approach

OSS projects and the value chain

Project categories
Code to product
Supporting market readiness

Community projects

Enterprise projects

Collaborative projects

Software is Code

```
72
73 def goTo(anActorReference, aPointToGoTo, aSpe
74     actorPosition = scenaric.GetPosition(anAc
75     if (type(aPointToGoTo).__name__ == 'strin
76         mark = scenaric.GetData(aPointToGoTo)
77         if (mark[0] == 1):
78             succeed = goToMark(anActorReferen
79         elif (mark[0] == 2):
80             succeed = goToPath(anActorReferen
81     elif (type(aPointToGoTo).__name__ == 'ver
82         succeed = goToVertex(anActorReference
83     else:
84         succeed = 0
85     return succeed
86
87
88
89 def goToMark(anActorReference, aMarkToGoTo, a
90     markPos = getMarkPosition(aMarkToGoTo)
91     succeed = goToVertex(anActorReference, ma
92     return succeed
93
94
95
96 def goToPath(anActorReference, aPathToFollow,
97     for coord in aPathToFollow:
98         astarPath = ge
99         if (len(astarP
100             for pos in
101                 succee
```

GitHub This repository Search Explore Features

RISCOSS / riscoss-platform-core

RISCOSS Open Source Risk Analysis Platform http://riscoss.eu/

23 commits 3 branches 0 releases

Branch: master riscoss-platform-core / +

lidix91 Update READ

etc

riscoss-platform-dm

可信的国家软件资

首页 项目分类 协同开发平台

常用链接

Trustie2.0 公共贴吧
Trustie2.0 开源软件趋势分析平台(ossean)
Trustie2.0 资源库
Trustie2.0 Git用户使用手册
Trustie大规模项目托管平台
Trustie大规模课程实践平台
软件资源库
TTA & NASAC 2010
技术发布与服务推广专题
Trustie是什么?
SoTrust 2011
项目中期考核

活动专题

主页 项目托管 公共贴吧 登录 注册

Trustie在线项目托管平台，面向中国大学生与软件从业者，提供社交化的项目管理、代码托管、资源共享、合作交流。

请输入要搜索的关键字 项目 搜索

热门项目 更多>>

Trustie-Forge (44人)
SocialForge是TrustieForge的升级版，支持社交化协同开发与合作。 SocialForge aims to provide a socialized and crowd so...
项目评分:20712

A better sandbox for BtrPlace (Entropy) (1人)
Btrplace is the flexible VM placement algorithm inside Entropy. It allows to compute the VMs plac...
项目评分:1036

Bench4Q (4人)
Bench4Q is a Cloud-Based software
项目评分:616

Trustie-OSSCEAN (8人)
Monitoring and mining open source software community data. This project uses webmagic framework, ...
项目评分:517

R-Memcached for Service4All (RMCS) (4人)

用户动态 更多>>

z9hang 更新了issue-edit: 缺陷 #1062 (已解决): 测试版--课程--课程...
更新于大约1小时前

z9hang 更新了issue-edit: 缺陷 #1069 (已解决): 在线测试版--项目--...
更新于大约1小时前

gyiang 更新了issue-edit: 任务 #78 (正在解决): OSS数据库转储
更新于大约 2 小时前

gyiang 更新了问题说明: 任务 #70: 管理缺陷
更新于大约 2 小时前

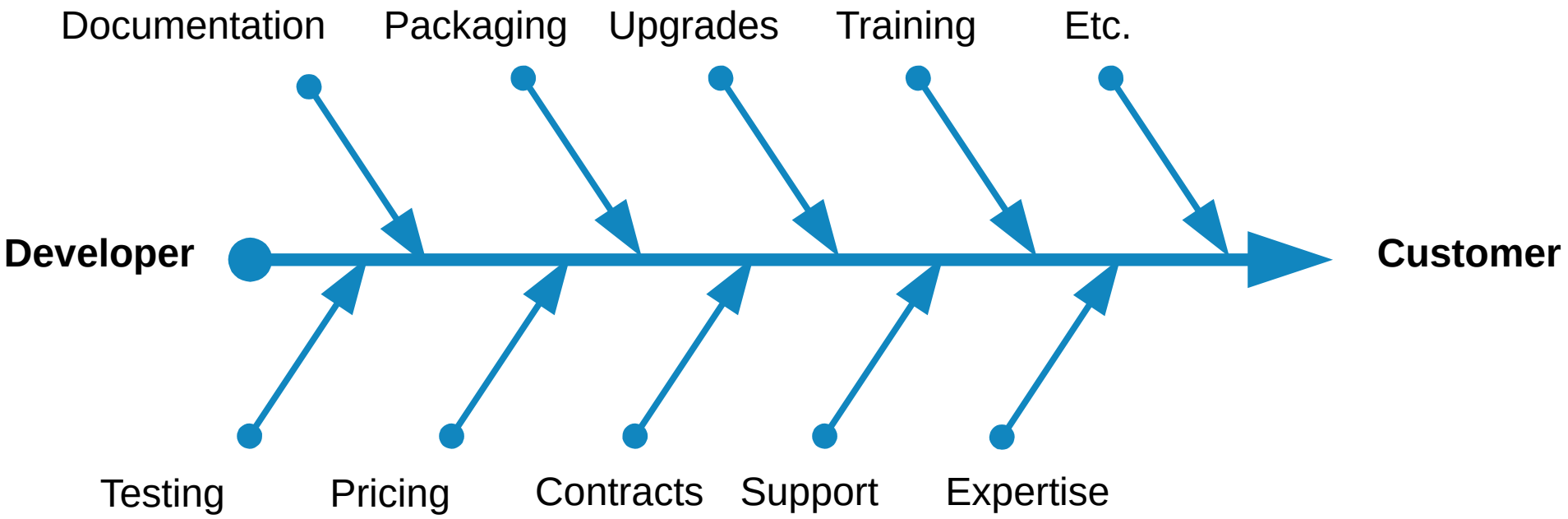
gyiang 更新了issue-closed: 任务 #38 (关闭): 将分散的表整理到一个数据库中
更新于大约 2 小时前

gyiang 更新了issue-edit: 缺陷 #28 (已解决): HackerNews的ur...
更新于大约 2 小时前

OW2 2017 Programming Contest 设

What is a Software *Product*?

What makes a Software *Product*?



What creates value?

Market Value

Code
POCs
Use-cases
Demonstrators

Documentation
Testing
Upgrades
Bug-fixing
Training
Support
Packaging
Case studies
Collateral
Pricing
Contracts
Early adopters
Etc.

Predictability
Quality
Trust

Research & Development

Delivery Challenge

OSS

Who creates value?
The ecosystem

Market Value

Code
POCs
Use-cases
Demonstrators

Documentation
Roadmap
Upgrades
Bug-fixing
Training
Support
Packaging
Case studies
Collateral
Pricing
Contracts
Early adopters
Etc.

Predictability
Quality
Trust

Systems Integrators

Open Source Orgs.

Contributors

Distrib. Vendors

Fiduciary Services

Users

Supporting market readiness and value creation

Market Value

Code
POCs
Use-cases
Demonstrators

Documentation
Roadmap
Upgrades
Bug-fixing
Training
Support
Packaging
Case studies
Collateral
Pricing
Contracts
Early adopters
Etc.

Predictability
Quality
Trust

Communication, Outreach, Marketplace

Governance, Projects, Initiatives, Quality Program

Collaborative Development Technical Resources

AppHub

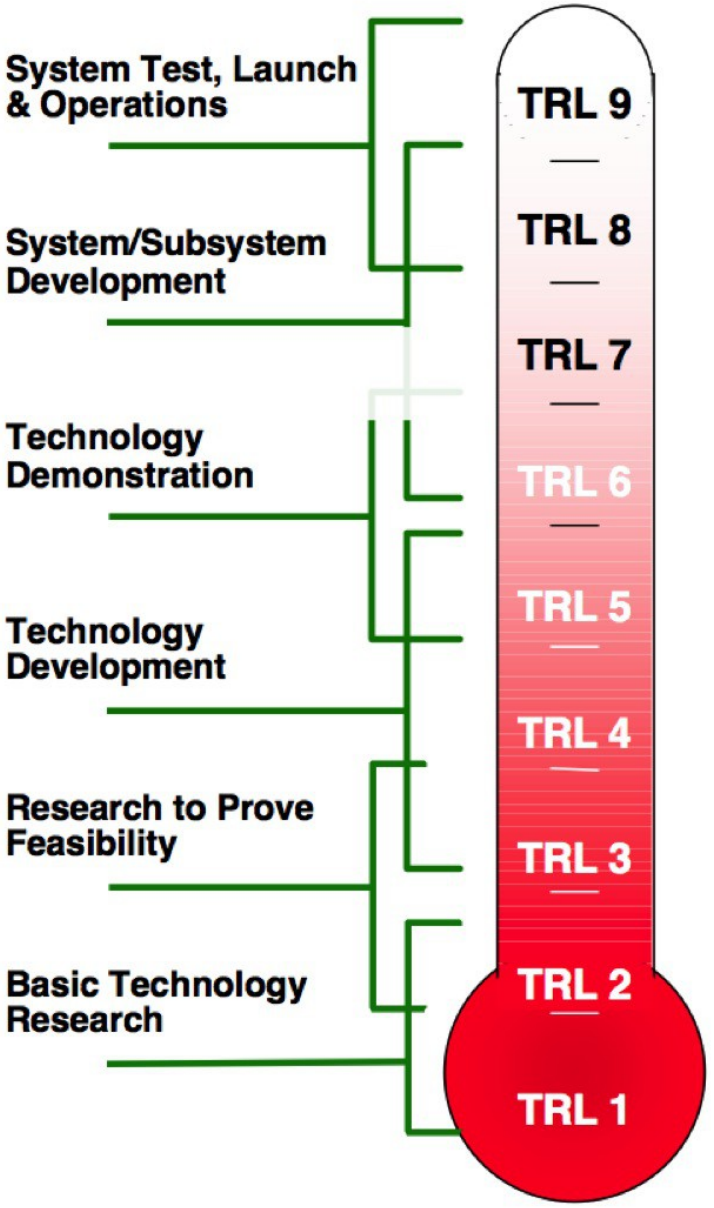
OSCAR

OW2Stack

Evaluating Readiness and Maturity

Technology Readiness Level
Market readiness
Open source readiness

NASA/DOD Technology Readiness Level



TRL 9 — Actual system "flight proven" through successful mission operations

TRL 8 — Actual system completed and "flight qualified" through test and demonstration (Ground or Flight)

TRL 7 — System prototype demonstration in a space environment

TRL 6 — System/subsystem model or prototype demonstration in a relevant environment (Ground or Space)

TRL 5 — Component and/or breadboard validation in relevant environment

TRL 4 — Component and/or breadboard validation in laboratory environment

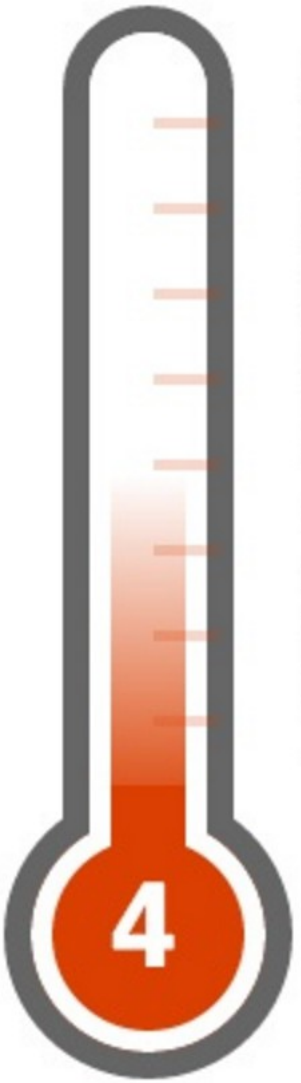
TRL 3 — Analytical and experimental critical function and/or characteristic proof-of-concept

TRL 2 — Technology concept and/or application formulated

TRL 1 — Basic principles observed and reported

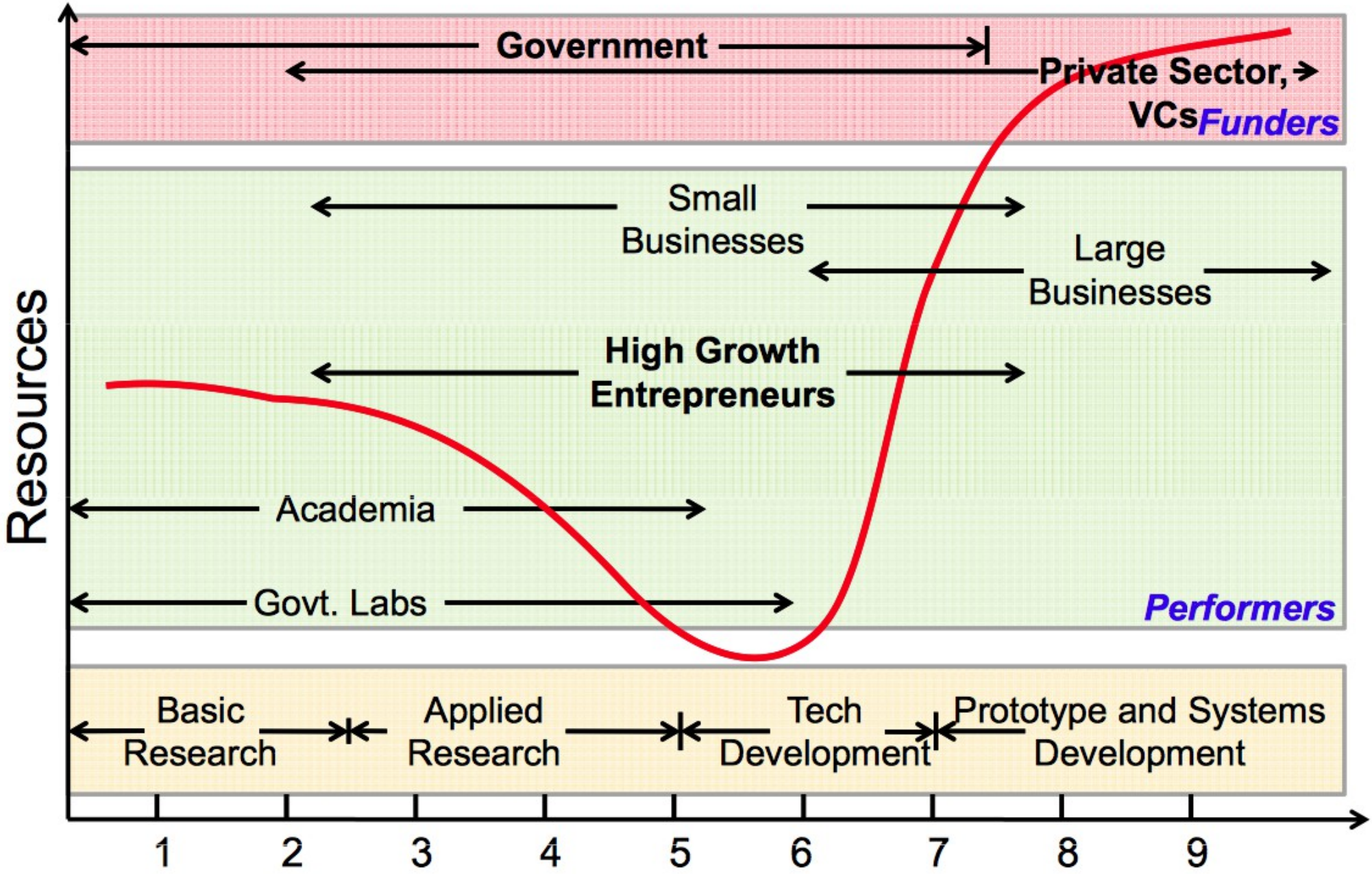
http://www.frankichamaki.com/wp-content/uploads/2014/01/nasa-trl.jpg

Investment Readiness Level



9. Validate Metrics That Matter
8. Validate Left Side of Canvas
7. Prototype High Fidelity MVP
6. Validate Right Side of Canvas
5. Validate Product/Market Fit
4. Prototype Low Fidelity MVP
3. Problem/Solution Validation
2. Mkt Size/Competitive Analysis
1. Complete First-Pass Canvas

https://steveblank.files.wordpress.com/2013/11/irl.jpg



Evaluating Open Source Projects

QualiPSo
OW2 SQuAT
CII Badge Program

OSS Analysis Landscape

Industry

Reuse Readiness Levels (RRLs)

EU Collab. Projects

Standardization bodies

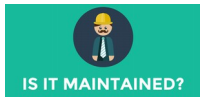
IP analysis

DejaCode

Qualitative analysis

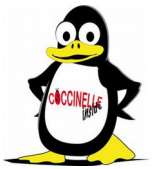
OW2 OMM forms

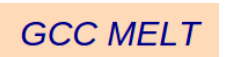
Engineering metrics



Bus Factor Calculator

Static analysis



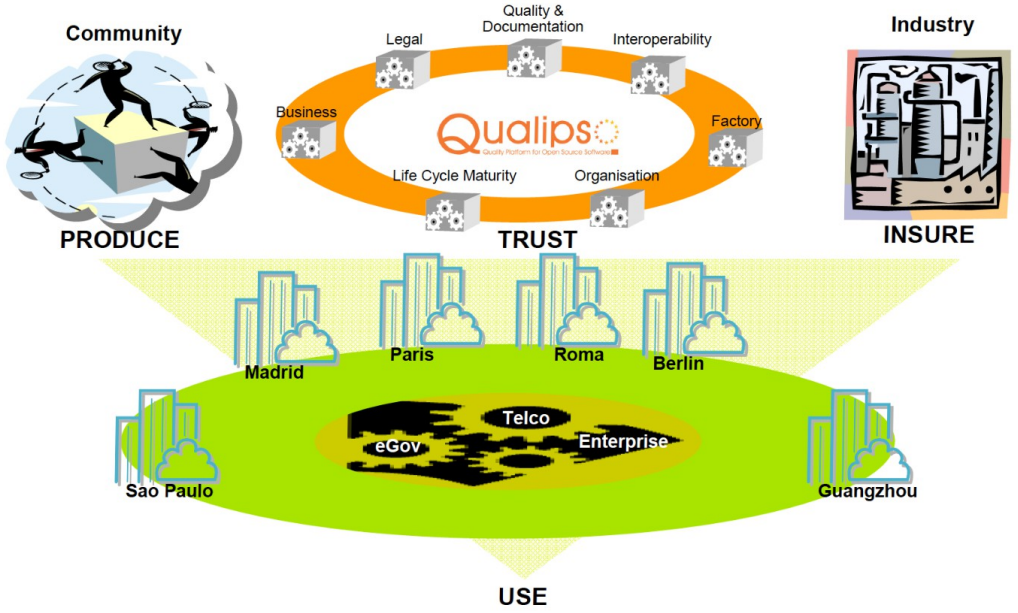


CI / Testing

Jenkins

Crowd testing

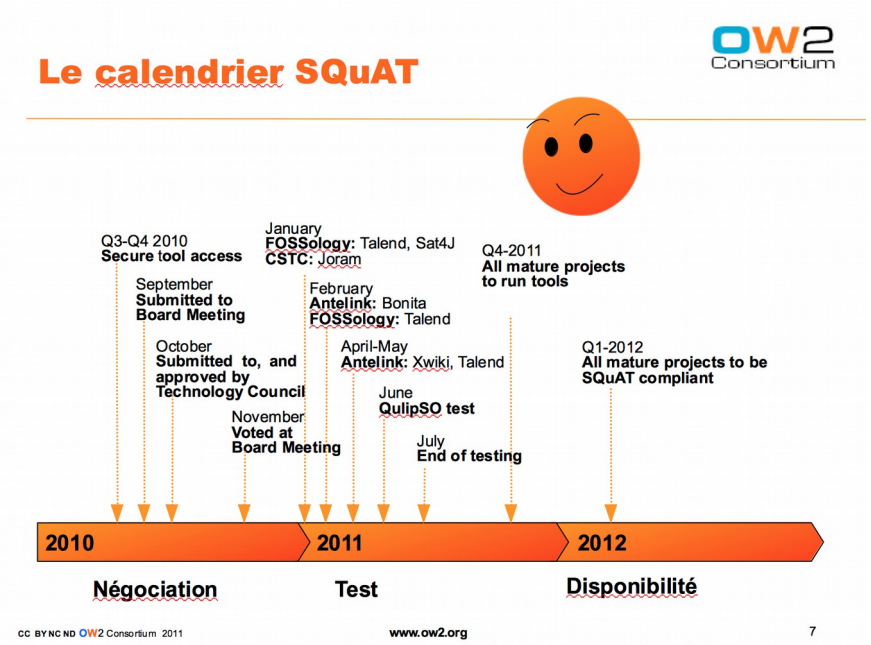
2007: QualiPSo



- European project
  - 48 months (2007-2010)
  - 22 organisations from 9 countries (3 continents)
- It is all about TRUST
  - Trust cannot be claimed without being proved!!!
- QualiPSo aimed at standardising the way OSS systems are built, offered and consumed.

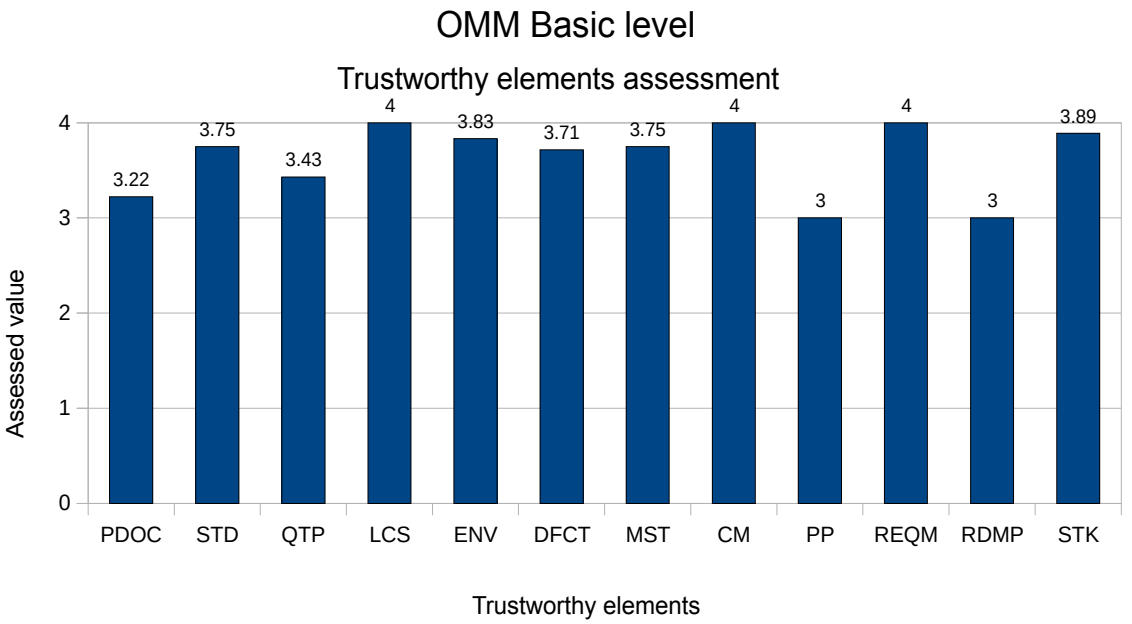
348		Moderate mailing lists / forums	0	1	2	3	4	
349		Moderate bug/issue management systems	0	1	2	3	4	
350		Survey regularly users what is their satisfaction level with available communication channels	0	1	2	3	4	
351		Adapt communication channels according to new requirements and comments provided by users	0	1	2	3	4	
352	Practice STK-2.2	Does the project measure regularly communication inside the community?		1	2	3	4	4
353	LookFor	Number of bug-tracking issues (evolution ratio)	0	1	2	3	4	
354		Number of bug-tracking issues unassigned (evolution ratio)	0	1	2	3	4	
355		Percentage of bug-tracking issues resolved (per week, per month etc)	0	1	2	3	4	
356		Number of writers in mailing lists (evolution ratio, classes: active, passive, quite active, very active)	0	1	2	3	4	
357		Number of mailing lists subscribers turned into developers (evolution ratio)	0	1	2	3	4	
358		Ratio of downloads this week over number of downloads in previous week	0	1	2	3	4	
359		Number of feature requests submitted by users/developers (evolution ratio)	0	1	2	3	4	
360		Number of bugs/issues submitted to the project (evolution ratio)	0	1	2	3	4	
361		Number of subscribers in the mailing lists (evolution ratio)	0	1	2	3	4	
362	Practice STK-2.3	Does the project measure the response rate inside different communication channels?		1	2	3	4	3
363	LookFor	Provide strong reactivity based on roles assignment	0	1	2	3	4	
364		Provide strong reactivity in mailing lists	0	1	2	3	4	
365		Provide strong reactivity in bug solution	0	1	2	3	4	
366		Provide strong reactivity in issues consideration	0	1	2	3	4	
367								
368	Goal STK 3	Does the project improve stakeholders involvement?						
369	Practice STK-3.1	Does the project measure the response level inside different communication channels and propose improvements?		1	2	3	4	4
370	Practice STK-3.2	Does the project improve the management style inside the project?		1	2	3	4	4
371	LookFor	Regularly assign roles related to different communication channels	0	1	2	3	4	
372		Regularly evaluate the quality of the communication inside the project	0	1	2	3	4	
373	Practice STK-3.3	Does the project improve the communication level inside the community?		1	2	3	4	4
374	LookFor	Take actions to prevent flaming	0	1	2	3	4	
375		Define communication rules	0	1	2	3	4	
376		Assign responsibility related to abuse inside the communication channel	0	1	2	3	4	
377		inappropriate communication (flaming, etc.) leads to loss of privileges	0	1	2	3	4	
378		Require from users that want to start actively communications inside the community to explicitly agree with defined communication rules	0	1	2	3	4	
379								
380								
381	**OMM: Basic Level**							
382	**Purpose: Check if the level is reached. If the percentage of fulfilment of practices is higher then 90 (percent) the Basic OMM level is reached.**							90.09
383								

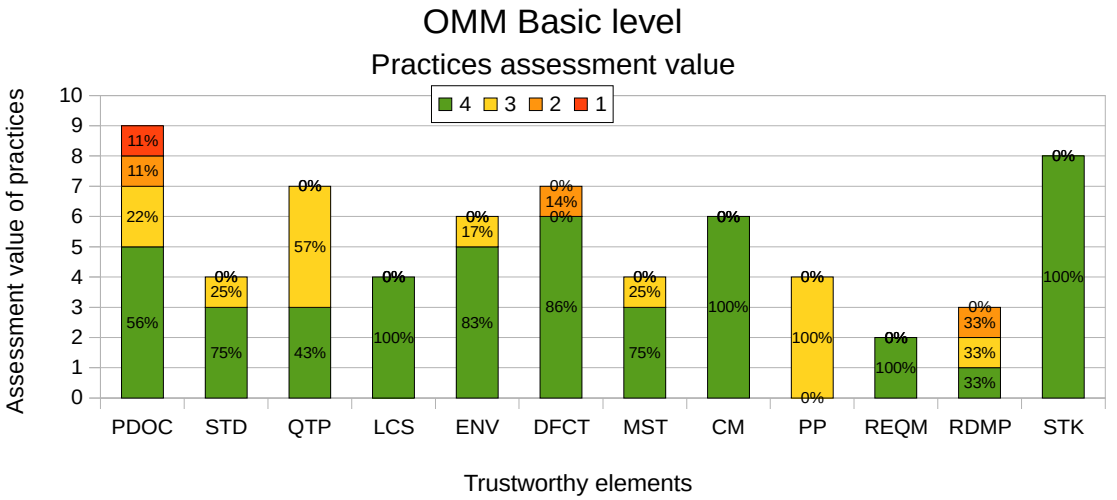
2010: OW2 Software Quality Assurance and Trustworthiness (SQuAT)



- IP verification: **FOSSology**
  - Applied on all OW2 mature projects
- Code verification: **Antelink**
  - Provides traceability of external libraries
- Static analysis: **Sonar**
  - Set of OW2 Sonar rules
- Maturity analysis: **Qualipso**
  - OMM applied to OW2 projects

OW2 Implementation of the QualiPSo OMM





2012: RISCOSS

???

Business Users Integrators

Commercial Products/Services

fossology

Ohloh

OSSMETER
Automated Measurement and Analysis of Open Source Software

markos

Antepedia

riscoss
MANAGING RISK AND COST IN OPEN SOURCE SOFTWARE PROJECTS

- Open source as a public resource freely accessible
- But OSS come from very different backgrounds
- Exploring and mapping the open source landscape
- Need to identify, measure, evaluate existing software
- Many tools and online services available

2015: CII Badge Program (Linux Foundation)

David A. Wheeler at OW2con'16

- Core Infrastructure Initiative (CII)
- Launched after the Heartbleed failure
- Organized by The Linux Foundation
- Supported by Amazon Web Services, Adobe, Bloomberg, Cisco, Dell, Facebook, Fujitsu, Google, Hitachi, HP, Huawei, IBM, Intel, Microsoft, NetApp, NEC, Qualcomm, RackSpace, salesforce.com, and VMware

LINUX FOUNDATION COLLABORATIVE PROJECTS

CORE INFRASTRUCTURE INITIATIVE

NEWS GRANTS PROGRAMS RESOURCES FAQ CONTACT

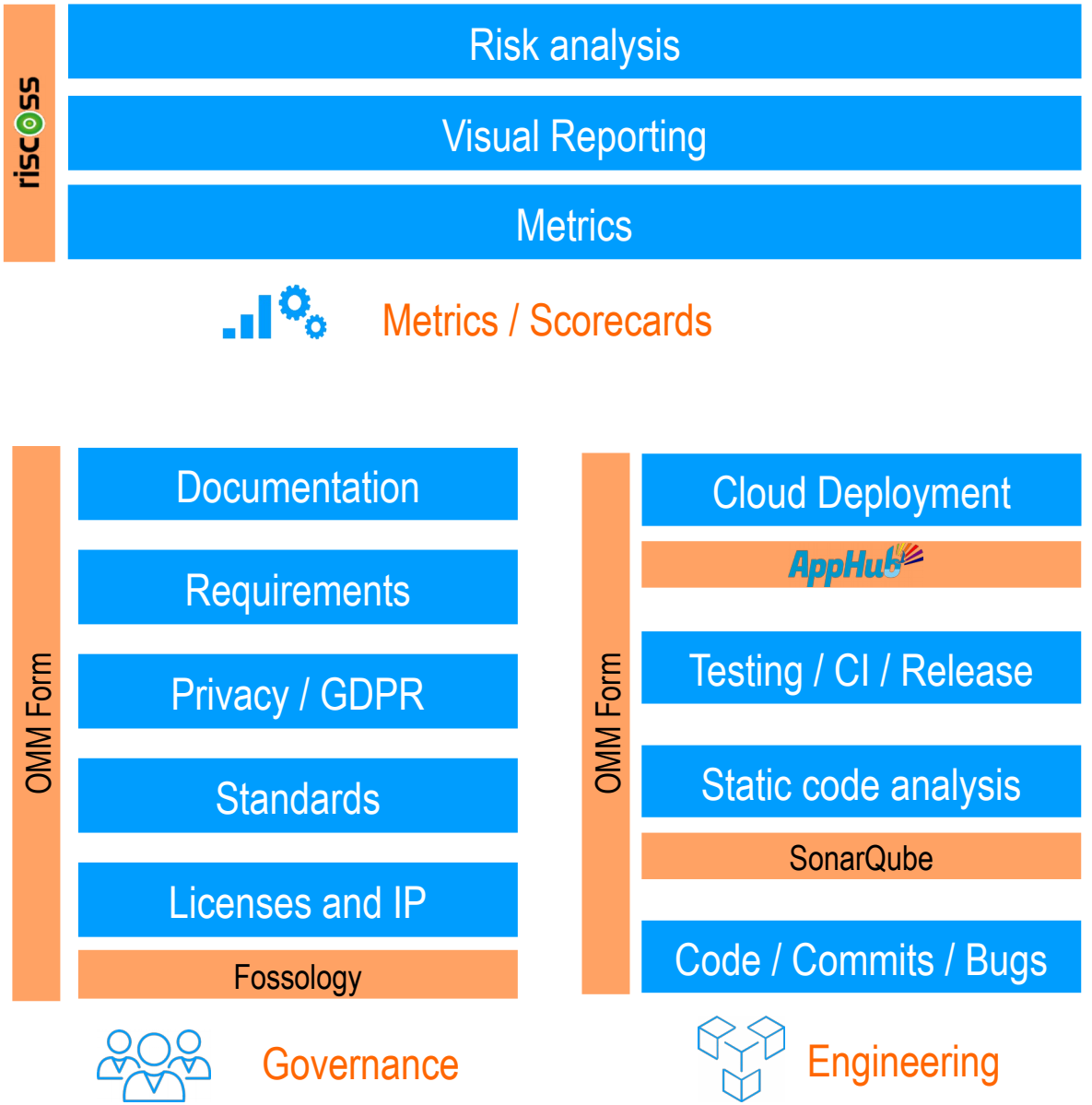
- https://www.coreinfrastructure.org/

CII BADGE PROGRAM	*Checklist:*
Basics	Project website
	Project website content
	FLOSS License
	Documentation
	Other
Change control	Public version-controlled source repository
	Version numbering
	Release notes (ChangeLog)
Reporting	Bug reporting process
	Vulnerability reporting process
Quality	Working build system
	Automated test suite
	New functionality testing
	Warning flags
Security	Secure development knowledge
	Good cryptographic practices
	Secured delivery mechanism
	Publicly-known vulnerabilities fixed
Analysis	Static code analysis
	Dynamic analysis

OW2 OSCAR Approach

More than just TRL
Market readiness
Promotes best practices

OSCAR Open Source Capability Assessment Radar



More to come:
- Accessibility
- Deployability
- Marketing
- Funding

OMM Assessment Web Form

ProActive OMM

Project Documentation

Purpose: Develop and maintain project documentation, making it readily accessible to the community

Asset documentation

Documentation related to assets need to be provided and properly maintained. Access needs to be provided for the community and relevant stakeholders.

PDOC-3.3	Does the project improve intrinsic quality of documentation?	75-100 % implemented
PDOC-3.4	Improve documents based on feedback and on evaluation	50-75 % implemented

Use of Established and Widespread Standards

Cloud Deployment

Purpose: assess cloud deployment readiness throuh quality requirements that have to be met in order to ensure that a project delivers deployable assets (binaries, images, templates) within a cloud marketplace such as AppHub.

DEP-1.1	Does the project provide cloud templates or images that are ready for deployment?	% implemented	75-100 % implemented
DEP-1.2	Does the project provide deployable templates / images versions that are aligned with the project's releases?	% implemented	75-100 % implemented
DEP-1.3	Does the project define a procedure for managing the deployable templates / images (e.g. as part of the configuration / release management)?	% implemented	50-75 % implemented
DEP-2.1	Does the project provide evidence that the deployable assets function correctly?	% implemented	25-50 % implemented

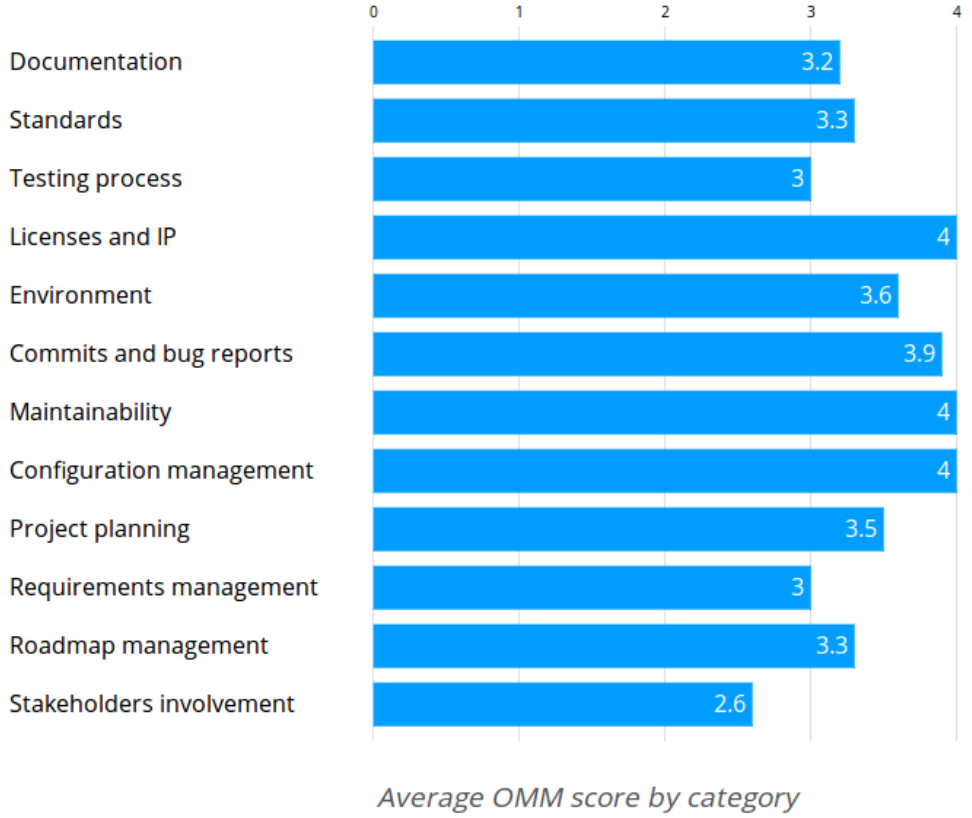
OMM Assessment

ProActive Parallel Suite

ProActive Workflows & Scheduling

Overview | Quality Assessment Tools | Risk Drivers | Market Readiness

FOSSology	upload=103&folder=11&item=413775
SonarQube	proactive parallel suite
OpenHub	ProActive
Open Maturity Model	ProActive Parallel Suite OMM



Average OMM score by category

FOSSology License analysis

ProActive Parallel Suite

ProActive Workflows & Scheduling

Overview | Quality Assessment Tools | Risk Drivers | Market Readiness

Licensing | Code Quality | Activeness

License	File count	Source
AGPL-3.0	940	FOSSology
No_license_found	427	FOSSology
MPL	12	FOSSology
Microsoft-possibility	6	FOSSology
BSD	6	FOSSology
Apache-2.0	5	FOSSology
MIT	4	FOSSology
GPL-3.0	4	FOSSology
OFL-1.1	2	FOSSology
MIT-style	2	FOSSology
MIT-possibility	2	FOSSology
GPL-possibility	2	FOSSology
GPL-2.0	2	FOSSology

SonarQube Static code analysis

ProActive Parallel Suite

ProActive Workflows & Scheduling

Overview | Quality Assessment Tools | Risk Drivers | Market Readiness

Licensing | Code Quality | Activeness

Metric	Value	Source
Lines	186,764	SonarQube
Ncloc	99,115	SonarQube
Classes	1,343	SonarQube
Files	1,133	SonarQube
Directories	247	SonarQube
Functions	7,539	SonarQube
Accessors	1,159	SonarQube
Statements	43,171	SonarQube
Public API	5,768	SonarQube
Comment lines	18,293	SonarQube
Comment lines density	15.6%	SonarQube
Public documented API density	46.1%	SonarQube

Risk Analysis

Risk Models

OW2 Risk Model

Description

The OW2 risk model is the risk model used by default on the OW2 projects' dashboards. Its normalization intervals are slightly more severe than the ones used by the basic risk model. It is less demanding in terms of license checking than the strict-IP model though.

Normalization intervals

unique-licenses	[0, 5, 8, 10, 20, Infinity]
ratio-no-license	[0, 5, 30, 40, 60, Infinity]
omm-license	[0, 2, 4, 5, 6, Infinity]
blocker-issues	[0, 1, 3, 5, 10, Infinity]
critical-issues	[0, 5, 10, 30, 50, Infinity]
test-coverage	[0, 10, 20, 40, 50, 100]
test-success	
openhub-one-year-cont	
openhub-activity	
omm-pdoc	
omm-std	
omm-qtp	
omm-env	
omm-dfct	
omm-mst	
omm-cm	

License Risk Function

```
function computeLicenseRisk(uniqueLicenses, noLicenseCount, fileCount,
   return (
           normalize(uniqueLicenses, "unique-licenses")
         + normalize((noLicenseCount / fileCount)*100, "ratio-no-licen
         + normalize(ommLcs, "omm-license")
         ) / 3;
}
```

Quality Risk Function

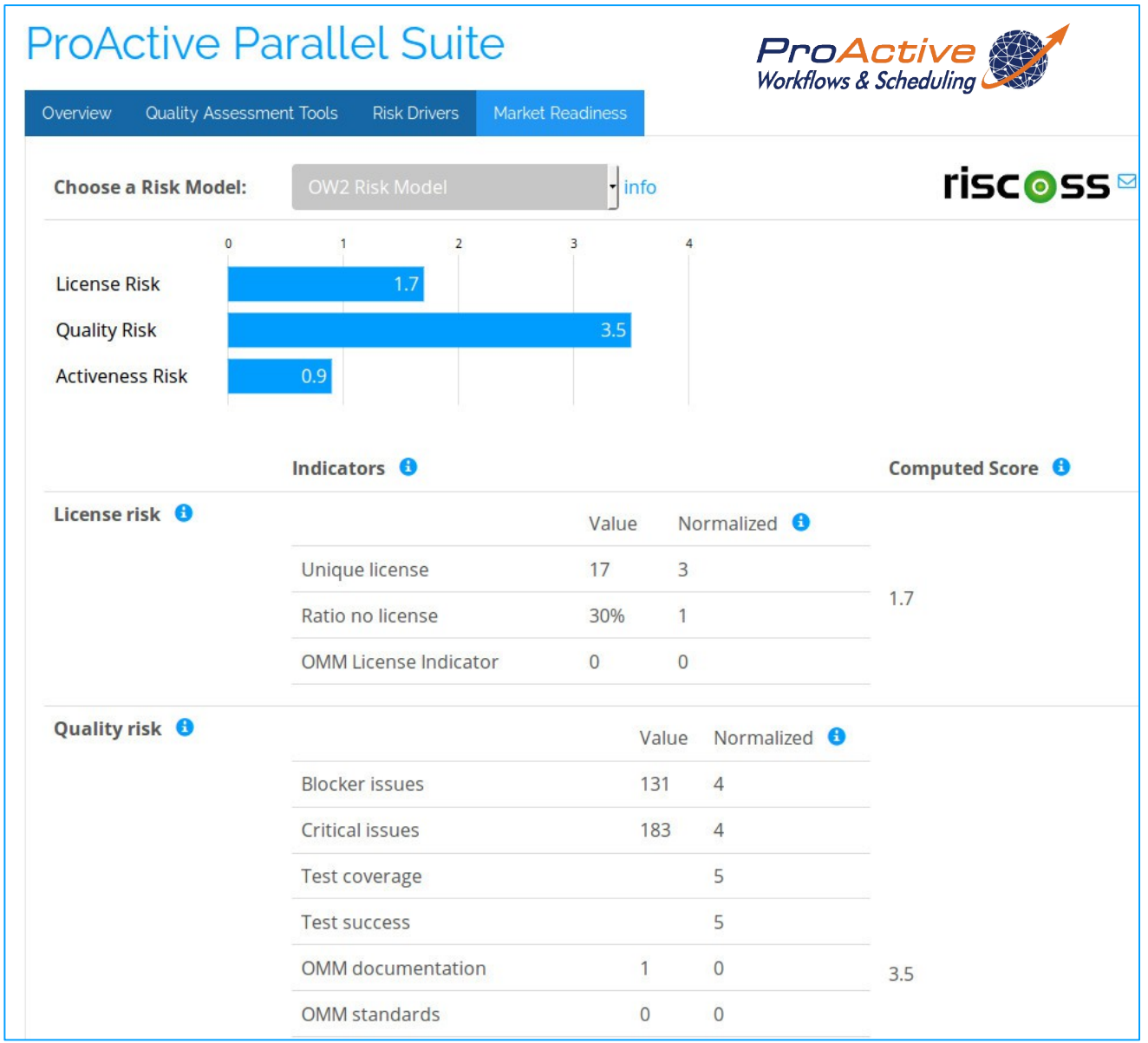
```
function computeCodeQualityRisk(blockerIssues, criticalIssues, testCov
   var ommAvg = (
             normalize(ommPdoc,"omm-pdoc")
           + normalize(ommStd, "omm-std")
           + normalize(ommQtp,"omm-qtp")
           + normalize(ommEnv,"omm-env")
           + normalize(ommDfct, "omm-dfct")
           + normalize(ommMst, "omm-mst")) / 6;

   return (
           (normalize(blockerIssues, "blocker-issues") + normalize(cri
         + normalize(testCoverage, "test-coverage", true)
         + normalize(testSuccess, "test-success", true) + ommAvg ) / 4
}
```

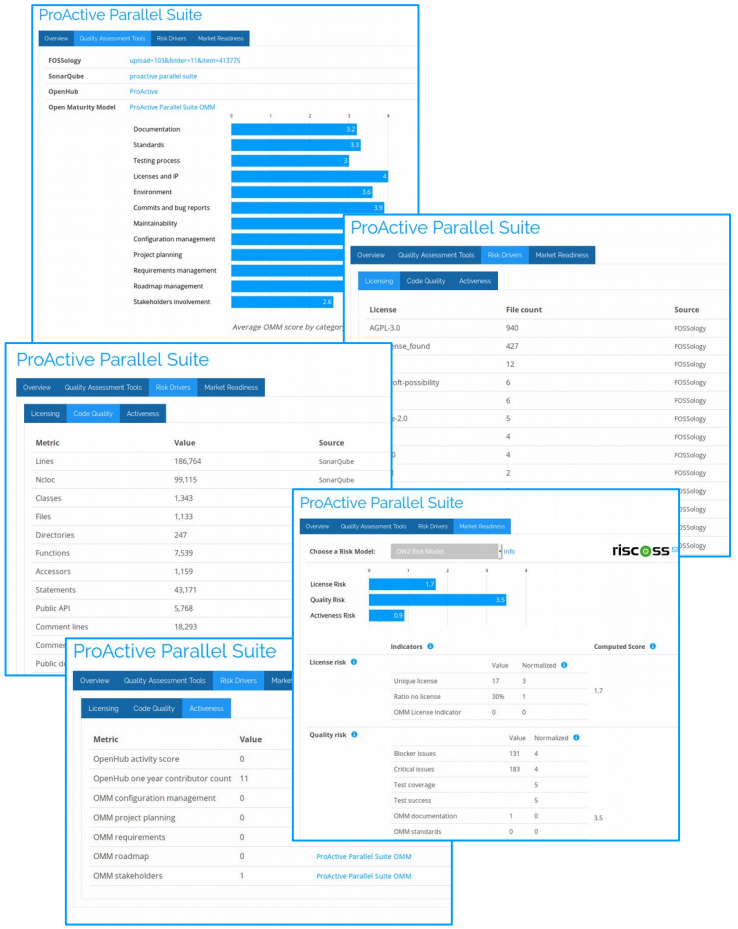
Activity Risk Function

```
function computeActivenessRisk(openHubActivity, openHubContributors, o
   return (
           normalize(openHubActivity, "openhub-activity", true)
         + normalize(openHubContributors, "openhub-one-year-contributo
         + normalize(ommCm,"omm-cm")
         + normalize(ommPp,"omm-pp")
```

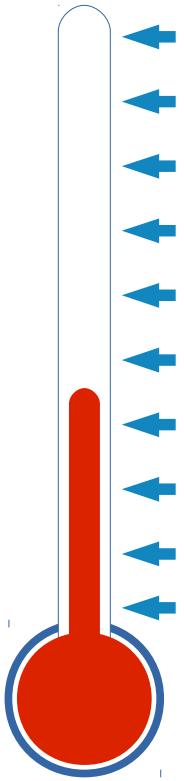
OSCAR
Market Readiness Scorecard



Work in Progress!



- Market Readiness Level
  - A type of measurement system used to estimate the market maturity and readyness of a particular project



Fully Supported Product (useful)

Rough Code (useless)

Thank You

And now let's talk
- **Q&A**
- **Disagreements**
- **Complements**
- **Feedback**
- **etc.**

www.ow2.org