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

Governance

Code Base

Non-Profit Open Source Organization
European and Global

Community

Activities

Members

Governance

Code Base

Non-Profit Open Source Organization
European and Global

Community

Activities

Members

Governance

Code Base

Non-Profit Open Source Organization
European and Global

Community

Activities

Members

Governance

Code Base

Non-Profit Open Source Organization
European and Global

Community

Activities

Members

Governance

Code Base
The freedom to run the software for any purpose
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
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 software
The freedom to run the software for any purpose

The freedom to study how the software works and to adapt it to your needs

The freedom to redistribute copies of the software

The freedom to improve the software and distribute 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

```python
def goTo(anActorReference, aPointToGoTo, aSpec
ctorPosition = scenario.getActorPosition(aActo
rReference)
if (type(aPointToGoTo) == "str"
    mark = scenario.getData(aPointToGoTo)
    if (mark[0] == 1):
        succeed = goToMark(anActorReference, aSpec
torPosition = aActorReference
    elif (mark[0] == 2):
        succeed = goToPath(anActorReference, aSpec
torPosition = aActorReference
    else:
        succeed = False
        return success

def goToMark(anActorReference, aPathToFollow, a
PathToFollow = aActorReference
    for cood in aPathToFollow:
        if (len(cood) > 0):
            for pos in cood:
                succeed = True
                return succeed
```

GitHub
```
RISCOSS / riscoss-platform-core

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

Branch: master  riscoss-platform-core / +
```

Trustie在线项目托管平台，面向中国大
学生和软件从业者，提供私密的项目管
理，代码托管，资源分享，交流合作。

用户动态
```html
Trus
tie Forge (4人)
Socialforge是Trus
tieForge的升级版，基于社会化协作开发平
台。Socialforge的目标是打造一个开放、
collaborative and socially aware platform for
the development of software.

BanchMQ (4人)
BanchMQ is a Cloud-based software development
platform.

Trustie OSS CS (8人)
Monitoring and mining open source software community dat
a. This project uses webmagic framework...
```

OW2 Programming Contest
What is a Software *Product*?
What makes a Software *Product*?

- Documentation
- Packaging
- Upgrades
- Training
- Etc.
- Developer
- Testing
- Pricing
- Contracts
- Support
- Expertise
- Customer
What creates value?

Without the code, the rest does not exist, but it's the rest that gives market value to the code.
Who creates value?
The ecosystem

Market Value

Contributors
- 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

Distrib. Vendors

Fiduciary Services

Open Source Orgs.

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
Evaluating Readiness and Maturity

Technology Readiness Level
Market readiness
Open source readiness
NASA/DOD Technology Readiness Level

- **TRL 1**: Basic principles observed and reported
- **TRL 2**: Technology concept and/or application formulated
- **TRL 3**: Analytical and experimental critical function and/or characteristic proof-of-concept
- **TRL 4**: Component and/or breadboard validation in laboratory environment
- **TRL 5**: System/subsystem model or prototype demonstration in a relevant environment (Ground or Space)
- **TRL 6**: System prototype demonstration in a space environment
- **TRL 7**: Actual system completed and “flight qualified” through test and demonstration (Ground or Flight)
- **TRL 8**: Actual system “flight proven” through successful mission operations
- **TRL 9**: System Test, Launch & Operations

**Research to Prove Feasibility**

**Technology Development**

**Technology Demonstration**

**System/Subsystem Development**

**System Test, Launch & Operations**
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
Evaluating Open Source Projects
OSS Analysis Landscape

Industry
- OSOS
- NASA
- Linux Foundation
- OW2
- BLACKDUCK | Open HUB

EU Collab. Projects
- Qualipso
- SQOLOSS
- OSSMETER
- FLOSSM3tr(cs)
- riscoss
- QualEdos

Standardization bodies
- SPICE
- ISO/IEC 15504
- CMMI

DejaCode
- IP analysis
- OW2 OMM forms

Qualitative analysis
- spago4Q
- Bitergia
- BLACKDUCK | Open HUB

Engineering metrics
- sonarqube
- GCC MELT

Static analysis
- Cppcheck
- mancoosi

CI/Testing
- Jenkins
- Crowd testing

2017, Cedric Thomas
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.
<table>
<thead>
<tr>
<th>Practice</th>
<th>STK-2.2</th>
<th>Does the project measure regularly communication inside the community?</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>LookFor</td>
<td></td>
<td>Nearest number of bug-tracking issues (evolution ratio)</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Percentage of bug-tracking issues resolved (per week, per month etc)</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Number of writers in mailing lists (evolution ratio, classes: active, passive, quite active, very active)</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Number of bugs/issuses submitted to the project (evolution ratio)</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nearest number of subscribers in the mailing lists (evolution ratio)</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Practice</td>
<td>STK-2.3</td>
<td>Does the project measure the response rate inside different communication channels?</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>LookFor</td>
<td></td>
<td>Provide strong reactivity based on roles assignment</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Provide strong reactivity in mailing lists</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Provide strong reactivity in bug solution</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Provide strong reactivity in issues consideration</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Practice</td>
<td>STK-3.1</td>
<td>Does the project measure the response level inside different communication channels and propose improvements?</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>LookFor</td>
<td></td>
<td>Nearest number of bug-tracking issues (evolution ratio)</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Percentage of bug-tracking issues resolved (per week, per month etc)</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Number of writers in mailing lists (evolution ratio, classes: active, passive, quite active, very active)</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Number of bugs/issuses submitted to the project (evolution ratio)</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nearest number of subscribers in the mailing lists (evolution ratio)</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Practice</td>
<td>STK-3.2</td>
<td>Does the project improve the management style inside the project?</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>LookFor</td>
<td></td>
<td>Regularly assign roles related to different communication channels</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Regularly evaluate the quality of the communication inside the project</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Practice</td>
<td>STK-3.3</td>
<td>Does the project improve the communication level inside the community?</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>LookFor</td>
<td></td>
<td>Take actions to prevent flaming</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Define communication rules</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Assign responsibility related to abuse inside the communication channel</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Inappropriate communication (flaming, etc.) leads to loss of privileges</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Require from users that want to start actively communications inside the community to explicitly agree with defined communication rules</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

| Practice | STK 3 | Does the project improve stakeholders involvement? | 1 | 2 | 3 | 4 | 4 |

**OMM: Basic Level**

**Purpose:** Check if the level is reached. If the percentage of fulfillment of practices is higher than 90 (percent) the Basic OMM level is reached.
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

OMM Basic level
Trustworthy elements assessment

OMM Basic level
Practices assessment value

2017, Cedric Thomas
2012: RISCOSS

- 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)

- 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

David A. Wheeler at OW2con'16
<table>
<thead>
<tr>
<th>CII BADGE PROGRAM</th>
<th>Checklist</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basics</td>
<td>Project website</td>
</tr>
<tr>
<td></td>
<td>Project website content</td>
</tr>
<tr>
<td></td>
<td>FLOSS License</td>
</tr>
<tr>
<td></td>
<td>Documentation</td>
</tr>
<tr>
<td></td>
<td>Other</td>
</tr>
<tr>
<td>Change control</td>
<td>Public version-controlled source repository</td>
</tr>
<tr>
<td></td>
<td>Version numbering</td>
</tr>
<tr>
<td></td>
<td>Release notes (ChangeLog)</td>
</tr>
<tr>
<td>Reporting</td>
<td>Bug reporting process</td>
</tr>
<tr>
<td></td>
<td>Vulnerability reporting process</td>
</tr>
<tr>
<td>Quality</td>
<td>Working build system</td>
</tr>
<tr>
<td></td>
<td>Automated test suite</td>
</tr>
<tr>
<td></td>
<td>New functionality testing</td>
</tr>
<tr>
<td></td>
<td>Warning flags</td>
</tr>
<tr>
<td>Security</td>
<td>Secure development knowledge</td>
</tr>
<tr>
<td></td>
<td>Good cryptographic practices</td>
</tr>
<tr>
<td></td>
<td>Secured delivery mechanism</td>
</tr>
<tr>
<td></td>
<td>Publicly-known vulnerabilities fixed</td>
</tr>
<tr>
<td>Analysis</td>
<td>Static code analysis</td>
</tr>
<tr>
<td></td>
<td>Dynamic analysis</td>
</tr>
</tbody>
</table>
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
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.

Cloud Deployment

Purpose: Assess cloud deployment readiness through 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.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEP-1.1</td>
<td>75-100 % implemented</td>
</tr>
<tr>
<td>DEP-1.2</td>
<td>75-100 % implemented</td>
</tr>
<tr>
<td>DEP-1.3</td>
<td>50-75 % implemented</td>
</tr>
<tr>
<td>DEP-2.1</td>
<td>75-100 % implemented</td>
</tr>
</tbody>
</table>

Use of Established and Widespread Standards

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
ProActive Parallel Suite

Overview | Quality Assessment Tools | Risk Drivers | Market Readiness
--- | --- | --- | ---
F OSSology | upload=103&folder=11&item=413775 |  |  |
SonarQube | proactive parallel suite |  |  |
OpenHub | ProActive |  |  |
Open Maturity Model | ProActive Parallel Suite OMM |  |  |
Documentation |  |  | 3.2 |
Standards |  |  | 3.3 |
Testing process |  |  | 3 |
Licenses and IP |  |  | 4 |
Environment |  |  | 3.6 |
Commits and bug reports |  |  | 3.9 |
Maintainability |  |  | 4 |
Configuration management |  |  | 4 |
Project planning |  |  | 3.5 |
Requirements management |  |  | 3 |
Roadmap management |  |  | 3.3 |
Stakeholders involvement |  |  | 2.6 |

Average OMM score by category
## FOSSology License analysis

### ProActive Parallel Suite

<table>
<thead>
<tr>
<th>License</th>
<th>File count</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGPL-3.0</td>
<td>940</td>
<td>FOSSology</td>
</tr>
<tr>
<td>No_license_found</td>
<td>427</td>
<td>FOSSology</td>
</tr>
<tr>
<td>MPL</td>
<td>12</td>
<td>FOSSology</td>
</tr>
<tr>
<td>Microsoft-possibility</td>
<td>6</td>
<td>FOSSology</td>
</tr>
<tr>
<td>BSD</td>
<td>6</td>
<td>FOSSology</td>
</tr>
<tr>
<td>Apache-2.0</td>
<td>5</td>
<td>FOSSology</td>
</tr>
<tr>
<td>MIT</td>
<td>4</td>
<td>FOSSology</td>
</tr>
<tr>
<td>GPL-3.0</td>
<td>4</td>
<td>FOSSology</td>
</tr>
<tr>
<td>OFL-1.1</td>
<td>2</td>
<td>FOSSology</td>
</tr>
<tr>
<td>MIT-style</td>
<td>2</td>
<td>FOSSology</td>
</tr>
<tr>
<td>MIT-possibility</td>
<td>2</td>
<td>FOSSology</td>
</tr>
<tr>
<td>GPL-possibility</td>
<td>2</td>
<td>FOSSology</td>
</tr>
<tr>
<td>GPL-2.0</td>
<td>2</td>
<td>FOSSology</td>
</tr>
</tbody>
</table>
### SonarQube Static code analysis

#### ProActive Parallel Suite

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lines</td>
<td>186,764</td>
<td>SonarQube</td>
</tr>
<tr>
<td>Ncloc</td>
<td>99,115</td>
<td>SonarQube</td>
</tr>
<tr>
<td>Classes</td>
<td>1,343</td>
<td>SonarQube</td>
</tr>
<tr>
<td>Files</td>
<td>1,133</td>
<td>SonarQube</td>
</tr>
<tr>
<td>Directories</td>
<td>247</td>
<td>SonarQube</td>
</tr>
<tr>
<td>Functions</td>
<td>7,539</td>
<td>SonarQube</td>
</tr>
<tr>
<td>Accessors</td>
<td>1,159</td>
<td>SonarQube</td>
</tr>
<tr>
<td>Statements</td>
<td>43,171</td>
<td>SonarQube</td>
</tr>
<tr>
<td>Public API</td>
<td>5,768</td>
<td>SonarQube</td>
</tr>
<tr>
<td>Comment lines</td>
<td>18,293</td>
<td>SonarQube</td>
</tr>
<tr>
<td>Comment lines density</td>
<td>15.6%</td>
<td>SonarQube</td>
</tr>
<tr>
<td>Public documented API density</td>
<td>46.1%</td>
<td>SonarQube</td>
</tr>
</tbody>
</table>
### Risk Analysis

#### ProActive Parallel Suite

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenHub activity score</td>
<td>0</td>
<td>OpenHub</td>
</tr>
<tr>
<td>OpenHub one year contributor count</td>
<td>11</td>
<td>OpenHub</td>
</tr>
<tr>
<td>OMM configuration management</td>
<td>0</td>
<td>ProActive Parallel Suite OMM</td>
</tr>
<tr>
<td>OMM project planning</td>
<td>0</td>
<td>ProActive Parallel Suite OMM</td>
</tr>
<tr>
<td>OMM requirements</td>
<td>0</td>
<td>ProActive Parallel Suite OMM</td>
</tr>
<tr>
<td>OMM roadmap</td>
<td>0</td>
<td>ProActive Parallel Suite OMM</td>
</tr>
<tr>
<td>OMM stakeholders</td>
<td>1</td>
<td>ProActive Parallel Suite OMM</td>
</tr>
</tbody>
</table>
# Risk Models

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.

<table>
<thead>
<tr>
<th>Description</th>
<th>Normalization intervals</th>
</tr>
</thead>
<tbody>
<tr>
<td>unique-licenses</td>
<td>[0, 5, 8, 10, 20, Infinity]</td>
</tr>
<tr>
<td>ratio-no-license</td>
<td>[0, 5, 30, 40, 60, Infinity]</td>
</tr>
<tr>
<td>omn-license</td>
<td>[0, 2, 4, 5, 6, Infinity]</td>
</tr>
<tr>
<td>blocker-issues</td>
<td>[0, 1, 3, 5, 10, Infinity]</td>
</tr>
<tr>
<td>critical-issues</td>
<td>[0, 5, 10, 30, 50, Infinity]</td>
</tr>
<tr>
<td>test-coverage</td>
<td>[0, 10, 20, 40, 50, 100]</td>
</tr>
<tr>
<td>test-success</td>
<td></td>
</tr>
<tr>
<td>openhub-one-year-contributors</td>
<td></td>
</tr>
<tr>
<td>openhub-activity</td>
<td></td>
</tr>
<tr>
<td>omn-pdoc</td>
<td></td>
</tr>
<tr>
<td>omn-std</td>
<td></td>
</tr>
<tr>
<td>omn-qtp</td>
<td></td>
</tr>
<tr>
<td>omn-env</td>
<td></td>
</tr>
<tr>
<td>omn-dfct</td>
<td></td>
</tr>
<tr>
<td>omn-mst</td>
<td></td>
</tr>
<tr>
<td>omn-cp</td>
<td></td>
</tr>
</tbody>
</table>

### License Risk Function

```javascript
function computeLicenseRisk(uniqueLicenses, noLicenseCount, fileCount, outLicenses, outNoLicenseCount, outFileCount) {
  var uniqueLicenses = normalize(uniqueLicenses, "unique-licenses");
  var noLicenseCount = normalize(noLicenseCount, "ratio-no-license");
  var fileCount = normalize(fileCount, "omn-license");
  return (uniqueLicenses + noLicenseCount + fileCount) / 3;
}
```

### Quality Risk Function

```javascript
function computeCodeQualityRisk(blockerIssues, criticalIssues, testCoverage) {
  var omxAvg = {
    normalize(omn-pdoc, "omn-pdoc")
    + normalize(omn-std, "omn-std")
    + normalize(omn-qtp, "omn-qtp")
    + normalize(omn-env, "omn-env")
    + normalize(omn-dfct, "omn-dfct")
    + normalize(omn-mst, "omn-mst") / 6;

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

### Activity Risk Function

```javascript
function computeActivityRisk(openHubActivity, openHubContributors) {
  return (normalize(openHubActivity, "openHubActivity", true) + normalize(openHubContributors, "openHubContributors", true) + normalize(openHubActiveCommits, "openHubActiveCommits", true) + normalize(openHubOpenIssues, "openHubOpenIssues", true) + normalize(openHubOpenIssues, "openHubOpenIssues", true)) / 5;
}
```
OSCAR
Market Readiness Scorecard

ProActive Parallel Suite

Choose a Risk Model: OW2 Risk Model

<table>
<thead>
<tr>
<th>Risk</th>
<th>Value</th>
<th>Normalized</th>
</tr>
</thead>
<tbody>
<tr>
<td>License Risk</td>
<td>1.7</td>
<td></td>
</tr>
<tr>
<td>Quality Risk</td>
<td>3.5</td>
<td></td>
</tr>
<tr>
<td>Activeness Risk</td>
<td>0.9</td>
<td></td>
</tr>
</tbody>
</table>

Indicators

License risk
- Unique license: 17, Normalized: 3
- Ratio no license: 30%, Normalized: 1
- OMM License Indicator: 0, Normalized: 0

Computed Score: 1.7

Quality risk
- Blocker issues: 131, Normalized: 4
- Critical issues: 183, Normalized: 4
- Test coverage: 5
- Test success: 5
- OMM documentation: 1, Normalized: 0
- OMM standards: 0, Normalized: 0

Computed Score: 3.5
Work in Progress!

Market Readiness Level

- A type of measurement system used to estimate the market maturity and readiness 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

For more details please contact Cedric Thomas, OW2 CEO, cedric.thomas@ow2.org