
Open Source Maturity Model: Moving OSS into the Mainstream

NAVICA

Copyright © 2006 Navica. All rights reserved.



© Scott Adams, Inc./Dist. by UFS, Inc.

What is Open Source Software?

- A licensing regimen**
- A collaborative development methodology**
- A way to reduce development costs**
- An inexpensive way to distribute software**
- An inexpensive way to procure software**
- A threat to proprietary software companies**



IDC: The biggest and most disruptive change to software in 25 years

Key Differences Between Commercial and Open Source Software

Commercial Software

- Restrictive license
- Bundled offering: explicit or implicit
- Active vendor, passive customer
- Key challenge: selecting right vendor

Open Source Software

- Expansive license
- Unbundled offering: due to OSS economics
- Active customer, passive vendor/project
- Key challenge: creating mature product

Clear need for open source evaluation methodology

Why are companies turning to open source?

Vendors

- Exhaustion of enterprise business model
- Time to market
- Cost
- Focus on core competency
- Reduce distribution/adoption time/cost/geographical constraints

End Users

- Cost
- Flexibility
- Time to market
- Try before you buy
- Lack of lock-in
- Opportunity for customization
- Collaboration with community

What do companies worry about?

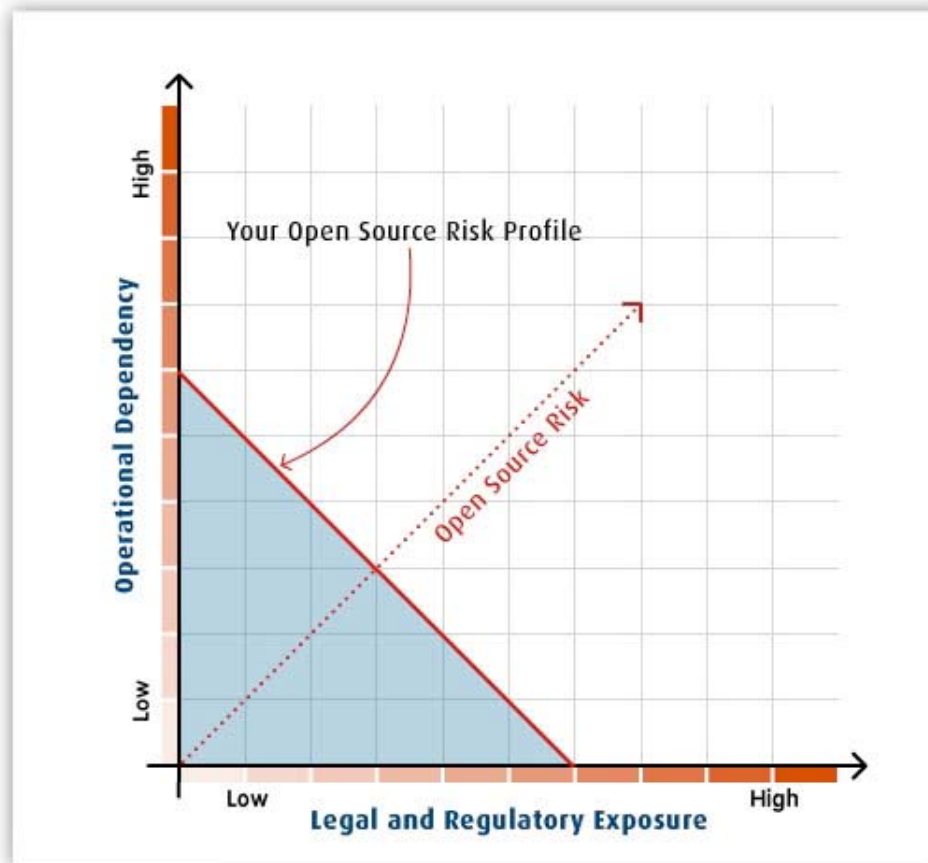
Vendors

- License compliance
- Viral licenses threaten business
- Product decisions made without technical and managerial review
- Ability to provide customer support
- Inconsistent decision-making process in organization

End Users

- License compliance
- Product proliferation
- Version proliferation
- High operational costs
- IT decisions made without technical and managerial review
- Inconsistent decision-making process in organization

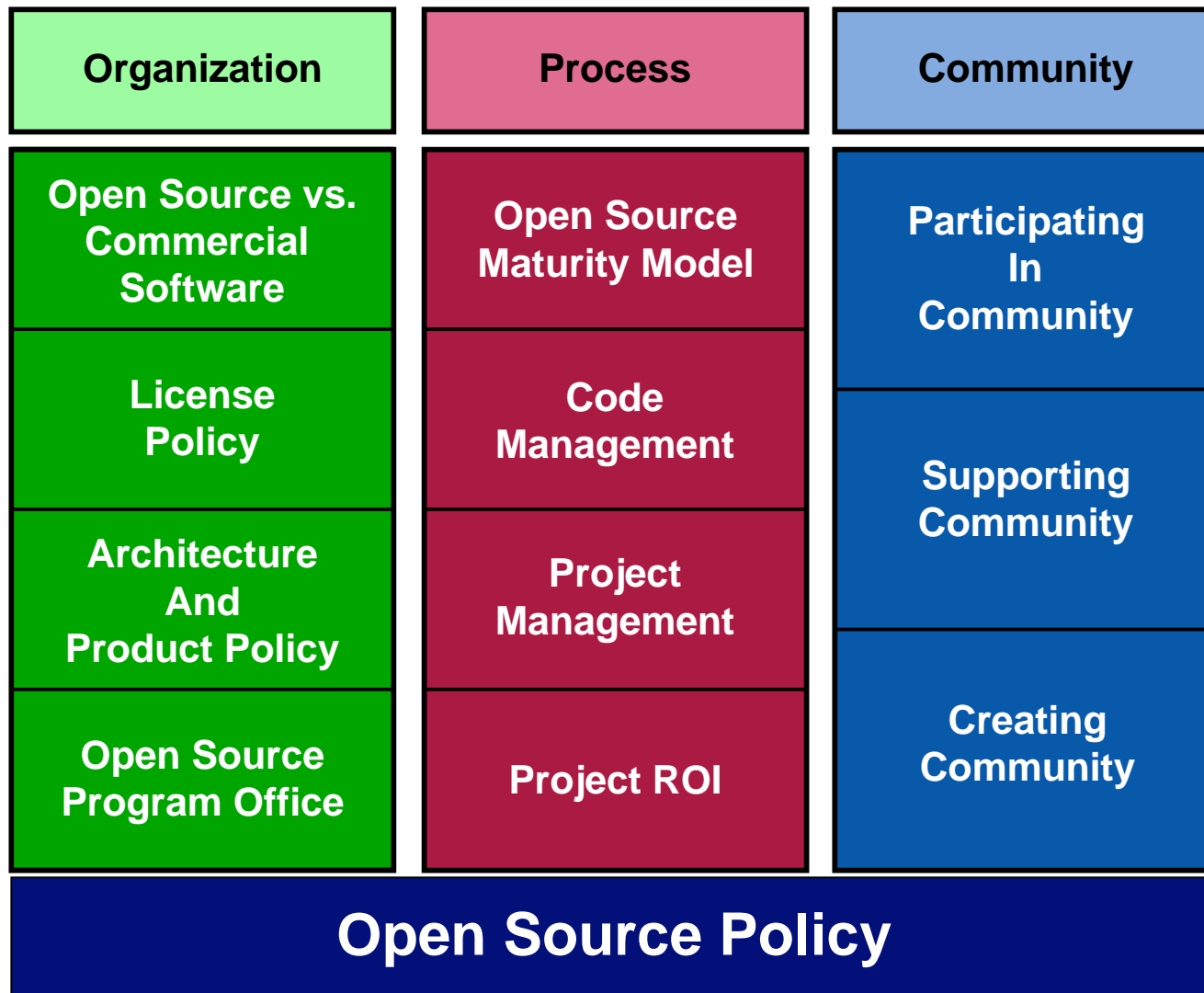
Open Source Risk Profile



Your Goal: To Imbalance Risk and Reward



Open Source Best Practices



Open Source Evaluation Requirements

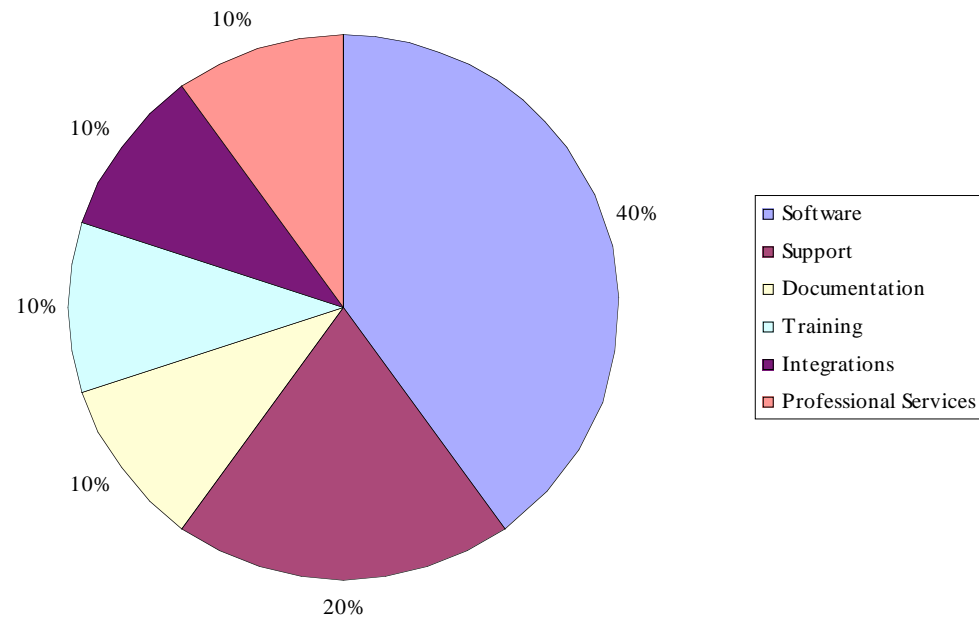
- **Must conform to categories already in use in organizations**
- **Must align with budgetary disciplines typically used by organizations**
- **Must be cost-effective and time-efficient for organizations**
- **Must offer clear-cut evaluation results**

Open Source Maturity Model

	Phase 1: Assess Element Maturity				Phase 2	Phase 3
	Define Requirements	Locate Resources	Assess Element Maturity	Assign Element Score	Assign Weighting Factor	Calculate Product Maturity Score
Product Software						
Support						
Documentation						
Training						
Product Integrations						
Professional Services						

Default Element Weightings

Software	4
Support	2
Documentation	1
Training	1
Integrations	1
Professional Services	1
Total	10



Recommended Minimum Maturity Scores

Purpose of Use	Type of User	
	Early Adopter	Pragmatist
Experimentation	25	40
Pilot /Departmental	40	60
Production	60	70

- **Recommended minimum scores vary according to use and organization type**

Conclusion

- **Open source represents profound change in software development, distribution, and implementation**
- **Current IT processes are not adequate to manage open source risk/reward**
- **Use of an evaluation methodology can reduce operational risk**