

Final Examination for the Course Opensource Software Paradigms

Maximum score: 50%

Time Allowed: 2hrs

Item	True/ False	MSQ	Short Answer	Scenario based	Total
Value	5%	25%	12%	8%	50%
Score					

Part I: True or False Items (5pts)

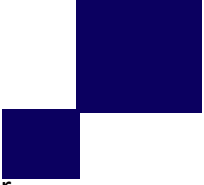
1. [False]
2. [True]
3. [True]
4. [True]
5. [True]

Part II: Multiple Choice Items (25pts)

1. [B]
2. [C]
3. [B]
4. [C]
5. [C]
6. [C]
7. [C]
8. [D]
9. [C]
10. [A]
11. [B]
12. [B]
13. [C]
14. [C]
15. [B]
16. [B]
17. [B]
18. [C]
19. [C]
20. [A]
21. [C]
22. [A]
23. [B]
24. [C]
25. [B]

Part III: Short Answer Items (12pts)

1. The "no warranty" clause is significant because it protects the original authors and contributors from legal liability. It explicitly states that the software is provided "as is" without any guarantees of reliability, merchantability, or fitness for a particular purpose. This is crucial as it shields developers from being sued for damages resulting from the use or failure of the software, making them more willing to share their work freely.
2. Contributor License Agreement (CLA): A formal agreement where a contributor grants the project the necessary intellectual property rights (like a license) to use their contribution. The contributor typically retains the copyright.



Copyright Assignment Agreement (CAA): A stronger agreement where the contributor actually transfers their copyright ownership of the contribution to the project's governing entity. This gives the project full control over the code.

Developer Certificate of Origin (DCO): A lighter, simpler alternative where a contributor certifies (by signing off on a Git commit) that they have the legal right to submit the code under the project's open source license. It does not involve a formal license grant or copyright transfer.

3. ODF is critical because it is an open standard, ensuring long-term access to documents without vendor lock-in. This guarantees that users can always access and edit their data using any software that supports the standard, protecting their digital sovereignty and preventing dependency on a single company's proprietary file format.
4. One key characteristic of the "Bazaar" development style is its decentralized and collaborative nature, where development is driven by a large, open community of contributors, leading to frequent releases and rapid evolution.
5. The primary goal of a Copyleft license is to ensure that the software, and all its modified versions or derivative works, remain free and open source, preserving the same freedoms for all downstream users.

Part IV – Scenario based Items (8pts)

1. License Change Analysis:

Changing to AGPLv3 would legally require MegaCloud to **release the source code of any modifications they have made to the CodeCraft tool** that they are running as part of their SaaS offering. The AGPL is specifically designed to close the "SaaS loophole" by treating network use as a form of distribution. This would force MegaCloud to either contribute back their improvements or stop using the newly licensed version of the software.

2. Business Model Proposal

Adopting an "Open Core" model would involve:

- Keeping the **core CodeCraft tool** open source (under the new AGPL license or another OSI-approved license).
- Developing a set of advanced, **proprietary "enterprise" features** (e.g., enhanced security, management dashboards, premium support, integration plugins).
- Selling licenses to these proprietary features to enterprises that need them, thereby generating direct revenue to fund the core team.

3. Community Impact

A potential risk is that the more restrictive **AGPL license could deter adoption** by other commercial companies or developers who are wary of the license's "virality" and obligations, even for internal use. Some might see it as less business-friendly than Apache 2.0, potentially slowing the project's growth and community contributions.