

Makara Ramoabi

20240045

IT Law

1. Scenario Analysis

An IT startup is developing a **web application** that integrates both proprietary code and open-source libraries. To ensure compliance, the development team must carefully select an appropriate license. The proprietary components, such as the unique user interface and branding, are protected under a **commercial license**, which restricts unauthorized distribution. However, the application also uses an **open-source encryption library licensed under GPL**. To comply, the developers must provide access to the source code of the integrated project, ensuring modifications and redistributions remain under the same license. Alternatively, they could replace the GPL library with an **MIT-licensed library**, which offers more flexibility by only requiring attribution. The licensing decision significantly impacts the startup's business model, as GPL promotes openness while MIT allows commercialization. By carefully analyzing license compatibility, the startup avoids legal conflicts, ensures ethical use of software, and builds trust with future customers and collaborators.

2. Concept Research

Open-source principles form the foundation of collaborative software development. They guarantee users the freedom to access, use, modify, and redistribute software under legally binding licenses. Key principles include **free redistribution**, **source code availability**, and the right to create **derived works**. These principles empower innovation by allowing developers to build upon existing projects rather than reinventing solutions. For example, the Linux operating system thrives under open source, powering everything from servers to smartphones. Open source also promotes transparency, as communities can inspect and improve software for security and performance. Ultimately, these principles foster innovation, collaboration, and cost-effective IT solutions.

3. Tool Practice

I reviewed a **GPL-licensed project on GitHub**, specifically focusing on how compliance requirements are documented. The repository included a **LICENSE file**, clearly outlining user rights and obligations, such as redistribution and source code disclosure. Contributors were required to credit the original author and publish derivative works under the same license. Reflecting on this, I learned that GitHub simplifies compliance by encouraging standard license

files and offering license selection tools when creating repositories. This practice not only prevents legal disputes but also builds trust in collaborative environments. Reviewing the GPL project reinforced the importance of documentation for legal clarity in IT.

4. Diagram Design

I created a **software licensing flowchart** in Draw.io to show how developers choose and apply licenses. The process begins with **project creation**, followed by a decision point: proprietary or open-source model. If proprietary, the flow continues with defining restrictions, pricing, and usage terms. If open-source, developers choose between **permissive licenses** (e.g., MIT, Apache) that require attribution or **copyleft licenses** (e.g., GPL) that require derivative works to remain open-source. After license selection, the next step is **compliance verification**, ensuring attribution and compatibility. Finally, the software is **distributed and monitored** for violations. The diagram clarifies the structured path to legal software release.



