70%

of modern components are OS code[1]

How do you know who to trust?

tiny

Tiny Technologies Inc.

Open Source Software Evaluation Checklist

How do you know who to trust?

Today, it's everywhere. Companies of all sizes – from small dev teams to large-scale enterprises – frequently use open source software. Why? Because when you're trying to innovate, fast, open source software (OSS) gives you a head start.

Open source is frequently the foundation of innovation across the software industry. And now, with the emergence of the tech assembly approach called 'digital factory', open source components are increasingly being used to facilitate faster project completions for both open and proprietary (or closed) applications. They're the outlier X-factor that, used cleverly, can positively impact growth and digital transformation.

The key is knowing which open source code to deploy, and in what way.

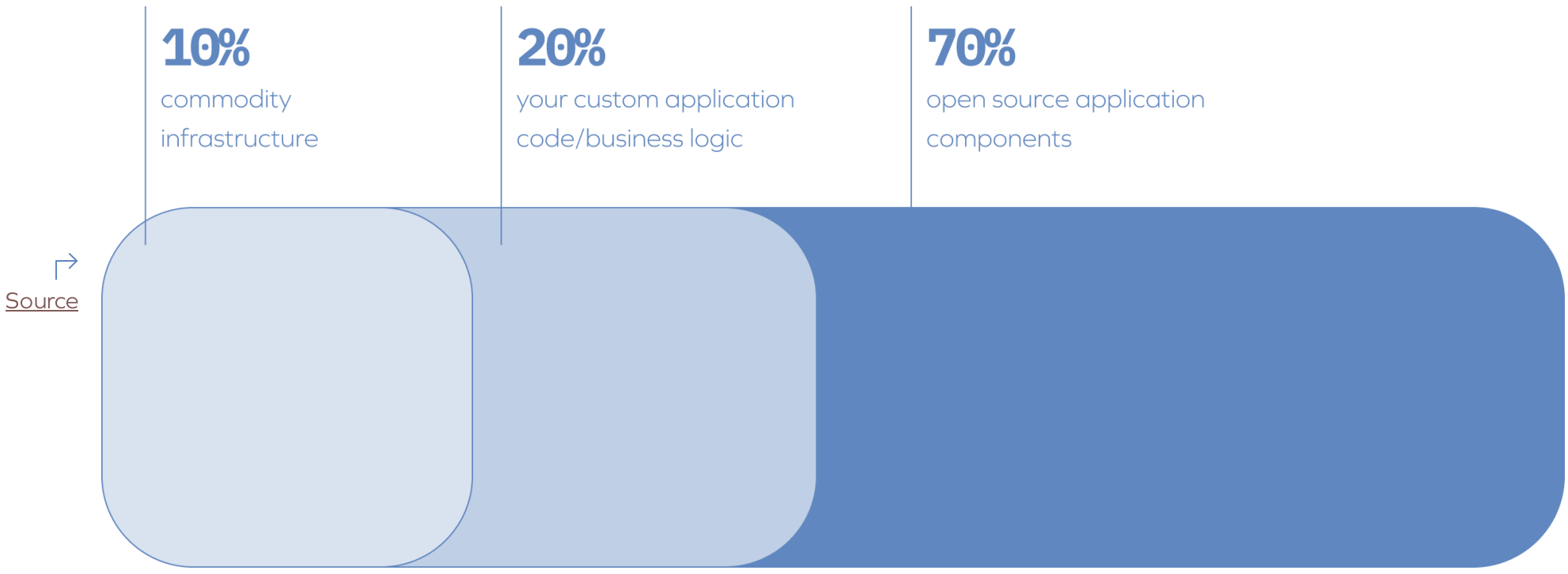
Having transformed from its once heretical roots, open source is now an essential factor in how enterprises evaluate, use and purchase software to drive their innovation plans. The rationale is simple: OSS lowers development costs, decreases time to market, increases developer productivity, and accelerates innovation.

What's a digital factory?

Research firm, McKinsey, has dubbed a factory-like tech assembly approach, a 'digital factory' – where a company "brings together the skills, processes, and inputs required to produce high-quality outputs. […] The best digital factories can put a new product or customer experience into production in as little as ten weeks. The innovation can then be introduced and scaled up across the business in eight to 12 months."

Why use open source software?

Billions of lines of code are freely available and shared through a community of creators, collaborators, and maintainers. As a result, most modern applications are built using open source components – in many cases, open source makes up more than 70% of the code.



Digging deeper, the reasoning for open source use is not dissimilar to other third-party SaaS or software purchases. Open source lets you leverage the velocity of your dev team talent – by letting them work on company differentiation instead of projects outside your core strategic focus.

It's a case of why reinvent the wheel, when others have already perfected it?

With a 'digital factory' approach, you assemble an agile software stack that comprises both open and closed components, which are curated, vetted, and professionally managed. It saves time, money and boosts your speed-to-market.

But how do you know who to trust?

Open source software evaluation criteria

In every development project, software evaluation is a critical piece of the puzzle – whether you're using open source or third-party closed components. It's a tricky balancing act between hard objectivity and subjective (but valid) individual user experience.

For open source software (OSS) components, the nine most often used evaluation criteria are:

- The number of developers working on the OSS
- The number of downloads of the software
- The developer's satisfaction
- The level of activity on the project
- The time between consequent releases
- The time to close bugs
- The security protocols in place
- The reputation in the community
- The quality control processes used

These open source specific criteria are normally added on top of the characteristics you already use, to measure closed (or proprietary) source software.

Assessment Criteria for Open Source Software

According to IGI-Global, "Since the code is available to everybody it can be reviewed and assessed by using traditional methodologies that measure the level of understanding, completeness, conciseness, portability, consistency, maintainability, testability, usability, reliability, structuredness and efficiency. These assessments can be done by everybody who is interested in the quality of the OSS."

Software assessment checklist: evaluating both open source and closed source components

Other areas to consider when evaluating both open and closed components:

- [ ] Is the development team familiar with the technology?
- [ ] Are large enterprises using the technology (ie. Fortune 500)?
- [ ] Does the technology provide features that do not currently exist?
- [ ] Does the technology have a strong capital backing (> USD$20M) or is it backed by a global corporation (e.g Facebook, Google, IBM, Apple)?
- [ ] At what speed is the technology growing?
- [ ] How much interest has been received from users?
- [ ] How much interest has been received from the community?
- [ ] How popular is the technology with the number of stars on GitHub (if applicable)?
- [ ] Is there a partnership that can be developed with the technology?
- [ ] Will this be an open source project?
- [ ] Does the technology allow us to provide features that do not currently exist?
- [ ] Have sales or existing users been lost, because of the lack of this technology?
- [ ] What type of feature (business model) will this be?
- [ ] How long will it take to build an MVP of the technology?
- [ ] How long will it take to build version 1.0 of the technology?

What’s a Software Bill of Materials (SBoM)?

According to Gartner, "SBoMs are an essential tool in your security and compliance toolbox. They help continuously verify software integrity and alert stakeholders to security vulnerabilities and policy violations.” They are an increasingly common and critical component of software development lifecycle (SDLC) and DevSecOps processes.

In 2021 several high-profile security breaches prompted the US President to issue an Executive Order on Cybersecurity, (May 2021). It included a requirement that software vendors provide a SBoM (Software Bill of Materials) for those selling to the US Federal Government. The order includes a provision that will require IT vendors to provide an SBOM with software and hardware.

Software evaluation process

Having an established process helps you minimize the unseen technical debt and code decay you're taking on with each component, while still allowing you to rapidly jump-start your innovation programs. Incorporate the following seven areas into that process:

- ☑ Create clear criteria to vet and pre-approve components (eg. Software Bill of Materials, SBoM)
- ☑ Ensure all components are monitored on an ongoing basis (eg. automated visibility and control of the components)
- ☑ Build a reusable tech stack composed of pre-approved and vetted components (eg. cataloging and building repositories)
- ☑ Have a clear understanding all licensing restrictions, requirements and opportunities
- ☑ Establish strong security protocols
- ☑ Ensure all components are regularly maintained and updated to QC standards
- ☑ Invest in service-level agreements (SLA) with all third-party component suppliers

By curating a repository of vetted, pre-approved components and release versions (open source, purchased and subscribed), you're minimizing possible security exposures as well as any ongoing maintenance and technical debt accumulation.

The components can then be safely used and reused within your tech stack, across the enterprise. That allows your devs to move more quickly and safely, and avoids last-minute blockers during a development cycle.

As noted in the Stripe Developer Coefficient Report, "…businesses need to better leverage their existing software engineering talent if they want to move faster, build new products, and tap into new and emerging trends."

Assembling pre-approved specialist components gives you the opportunity to safely walk the tightrope between technical debt and innovation.

And it maximizes your opportunity cost.

What's Managed Open Source Software?

Managed open source software and components, maximize the speed-to-market, while minimizing potential risks to users. All license details, security risks and maintenance issues are clearly communicated (if they occur) and upgrades are regularly released.

1 Tidelift