https://www.opentechstrategies.com/archetypes

Archetypes At A Glance

	Main Benefit	Main Drawback	Situational Considerations	Dev Speed	Typical Participants	Ease of Onboarding	Community Standards	Typical Governance	Measure of Open Source Success
B2B	Driving industry adoption of your technology.	Little or no collaborative development.	Requires major market power to be effective.	Fast; pace set by business goals.	Organizational reps.	Hard.	Oriented toward organizations.	Founding org, w/ some partner influence.	Adoption by target partners; successful projects built around core project.
Multi-Vendor Infra	Collaboration with partners; address a set of shared problems.	Sometimes off-putting to individual contributors.	Business needs of participants affect community management.	Usually moderate, but depends on needs of participants.	Organizational reps.	Medium.	Welcoming but formal; difficult for individuals.	Committee of organizational reps.	Partner org variety & participation; participant longevity.
Rocket Ship To Mars	Quick, focused effect in a specific area.	Collaboration only available from those who share a very specific vision.	Everything depends on success of original vision.	Fast; escape velocity.	Founding organization.	Hard.	Focused on core group.	Founder governs with iron fist.	Dev speed; adoption by target users; reaching tech goals.
Single-Maintainer Houseplant	Easy to start.	Single maintainer may become burdened by success.	Starts by filling a small niche, then grows.	Medium - fast.	Founding dev and one-off contributors.	Varies.	Varies.	Founder leads.	Either is stable with single maintainer or eventually transitions to another archetype.
Controlled Ecosystem	Can build a sustainable ecosystem in which founding organization has strong influence.	Compromise needed to avoid forks (esp. commercial).	Participants have many motivations (commercial & non-commercial).	Medium.	Founder, some external core contributors, many plugin contributors.	Medium.	Welcoming, with some onboarding structures.	Benevolent dictatorship; tries to avoid forks.	Adoption by target users; extension developers growth.
Wide Open	Large-scale collaboration; community can become self-sustaining.	Effort to maintain onboarding paths & manage all participants.	Differing commitment & engagement levels among participants.	Slow - medium; some process overhead.	Open to anyone; participant demographic depends on project.	Easy.	Very welcoming, formalized onboarding systems.	Group-based; consensus / democratic.	Contributor growth; contribution efficiency; variety in where ideas and decisions originate.
Mass Market	Large user base can make project broadly influential.	Huge user base needs filtering for dev community.	Contributor base does not accurately represent user base.	Slow - medium; swift change destabilizes user base.	Organizational reps, redistributor reps; some users who are technical.	Easy to medium.	Fully open; scales via users helping users.	Main organization leads, with outside input.	User awareness that product is FOSS; non-technical contributor growth; effective filtering of user feedback to devs.
Specialty Library	Ensure quality solution to a specific problem; can lead to new partners.	High barriers to entry; relatively small developer pool.	Standard-setting effects (*de facto* or official).	Gets slower over time, as library stabilizes.	Developers with expertise in the relevant field.	Depends on technical complexity.	High barrier; contributors need to be experts.	Multi-party committer group.	Adoption in intended domain; high quality of contributors and contributed code.
Trusted Vendor	Loyalty of downstream consumers helps project stability.	Primary org must be careful how it uses its position.	Customer needs vs open source project needs.	Medium. Primary vendor momentum vs third-party needs.	Customer reps (both paying and non-paying); some one-off contributors.	Medium to hard.	Clear boundaries: users have mainly roadmap input.	Main vendor leads.	Lack of competitive forks; vendor's leadership accepted by community.
Upstream dependency	Broad reach across (hence insight into) many dependee projects.	Developer base can sometimes be lightly motivated.	Usage patterns of downstream consumers.	Medium; may slow down as standard settles.	Downstream devs.	Depends on technical complexity.	Welcoming; amenable to one-time contributions.	Informal, maintainer-led, committer groups.	Multiple competitive uses; participant longevity; bug reports are technical and constructive.