Stewarding Your Open Source Project: A Guide

Last Updated: 1 October 2021

Audience

This guide is for practitioners with an open source project that **has or intends to have strong community participation**.

This guide is intended to help practitioners answer two questions:

- What open source strategy should I use to achieve a business goal?
- How do I best manage my project, especially the non-code aspects, to achieve my strategy?

This guide:

- walks you through each open source lifecycle phase
- explains what questions to consider and what steps to take in each phase
- provides links to templates and resources (they are being created over time)

Key principles and definitions in this guide are explained in the Appendix.

Out of scope

This guide is not for practitioners who open source example code, samples, demos, or personal projects, or projects that expect no community participation. This guide does not address the question: “should I create code or use an existing project?”

Table of Contents

Lifecycle Stage: Seed

About this stage

This is the stage where you consider open sourcing. You determine if and how open sourcing achieves a business goal. If it does, then you document your open source plan and gain stakeholder approval. This entails reviewing your plan with engineering and product management leadership for written sign off. This phase is also the time to understand what kind of investment (effort, resource, funding) are required overtime to achieve your project's goal.

Key Questions to Consider

These questions are designed to help you prepare for this stage.

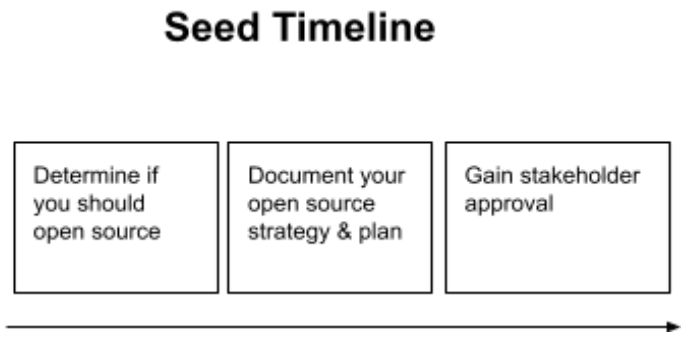
- What business goal will I achieve by open sourcing?
- What open source strategy best helps me achieve my business goal?

- Which leaders in your organization will be stakeholders for this open source project and approve the open source strategy?
- What are the metrics for success through the lifecycle of the project?
- What approach should I use to support the project within each open source pillar? (e.g. type of technical governance, how to grow community).
- What support will I need outside of my team in order to steward my project? What deadlines do I need to set for this support?
- What are the initial and long term investments needed to support the project's success?

Key Steps To Take

In this stage, be sure to complete the Must Do steps. For a recommendation of priority order, see the timeline below.

Focus Areas	Steps to Take
Strategy & Planning	
Must Do	• Write a plan that details the business goal and the open source strategy to achieve it • Determine your project's goals including a general game plan to address each pillar • Set success near-term and long-term metrics
Nice To Have	• Multi-year strategy and plan that anticipates growing through each phase of the lifecycle. The plan considers the full investment ranging from additional staff to marketing/PR efforts, events, etc. • Documentation outlining the technical concepts behind your open source project
Stakeholder Management	
Must Do	• Get approval to open source



Lifecycle Stage: Sprout

About this stage

If you have a clear business reason to open source and you have stakeholder approval, then you proceed to the Sprout stage, where you launch your project to the world!

Other key tasks in this stage include:

- Creating basic documentation
- Add a Code of Conduct for your project
- Decide on and create clear decision-making processes and policies, including:
    - technical and community leadership roles
    - clear paths for community members to contribute to your project
- Announcing your project on your company blog or social media channels
- Setting up communication and collaboration channels (e.g. Google Groups, Slack, Discord)

Key Questions to Consider

These questions are designed to help you prepare for this stage. Links to topical guides are provided in the section to help you answer the questions.

- **Release Management:** Do I know how to prepare for and go through my organization's release process? What is the release schedule?
- **Infrastructure:** Where will the project's main repository reside? How will code, pull requests, and issues be synced between internal and external repositories, if necessary?
- **Branding:** What is a good name for my project and how do I get it approved? How will I describe my project when I announce it to the world? How do I promote my project and make it findable within my organization and externally?
- **Governance:** What governance model is the best fit with the open source strategy now and in the future?
- **Community Development:**
    - What is my community plan?
    - Who do I want in my community (users, contributors, ecosystem partners)?
    - Are there any groups whose absence would cause my project to fail?
    - What is the ideal size of my community?
    - What is the ideal user: contributor ratio?
    - What are all the groups that would find the project useful; how can I reach them?
    - Will target companies contribute to the project or become ecosystem partners if my company owns the code base or does the code need to move to a foundation?

    - How do I build diversity into my community both at the leadership level and overall?
- **Contributor Experience**: Do I need or expect external contributions? What documentation can I provide to strengthen contributor onboarding? Am I providing 'good first issue' features or fixes to encourage new contributors?
- **Production readiness:** What documentation will users need to get started with this project? What are my plans for testing, security, accessibility, usability, vulnerability disclosure and CI/CD?
- **Stakeholder:** What do my stakeholders need to know about my efforts in this stage? Do they want to know when the project is released?

Key Steps To Take

In this stage, be sure to complete the Must Do steps. For a recommendation of priority order, see the Timeline below.

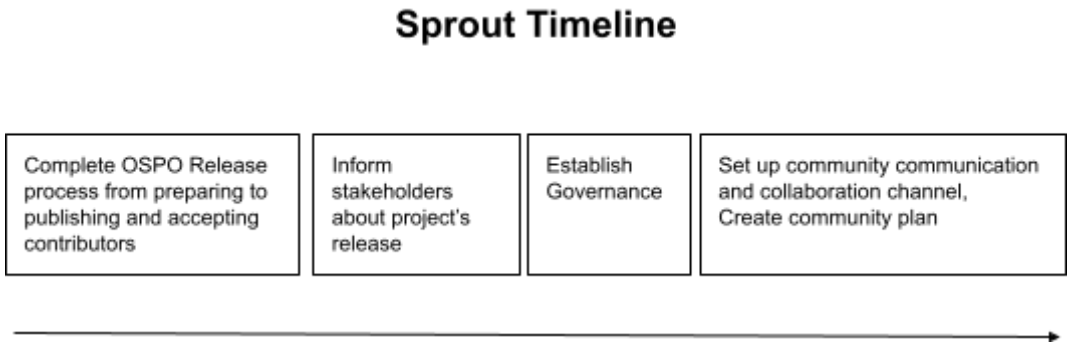
Focus Areas	Steps to Take
Branding	
Must Do	• Work with your organization's marketing team to develop your project name • Set up accounts to reserve namespaces in communications channels you hope to be active in (e.g., Twitter, website, YouTube, etc.) • Set up community communication channels such as Slack, Discord, Gitter, mailing lists, Google Groups
Nice to Have	• Create branded presentation templates to share with community members/contributors
Governance	
Must Do	• Determine your project governance • Create written policies, including a code of conduct enforcement policy, and make them available
Nice To Have	• Set up a special interest group (SIG) for code of conduct enforcement and document the enforcement process
Production Readiness	
Must Do	• Inline examples/test cases, either in README or another doc 📖 • User quickstart/getting started guide 📖 • Communications guidance (e.g. "we answer questions [here], you can expect responses in [X time period]" 📖 • Test documentation, including instructions for running tests 📖 • Communication channel for reporting vulnerabilities/CVE management • Accessibility plan • CI/CD

Nice To Have	• Tutorials 📖 • Changelog 📖 • Usability research • Security scanning
Release Management	

Must Do	• Document your release management process, including:
	  ○ Who manages the releases?
	  ○ What is the versioning policy?
	  ○ What are the testing requirements? How do we do release-specific testing?
	  ○ What are the release-blocking criteria and tests?
	  ○ How will we manage: patch management; cherry picks; rollbacks, roll-forwards, and reverts; vulnerability patch releases; branches, etc.?
	  ○ How do releases go from alpha to beta to GA?
	  ○ How will we ensure release cadence consistency?
	  ○ When will we do release retrospectives? Will they be open to the community?
	  ○ Do we have a deprecation process?

Nice To Have	• Publish a basic, public roadmap 📖
Community Development	
Must Do	• Announce your project and ask your team and influencers to amplify the announcement • Create a community plan defining what types of members you want in terms of users, ecosystem partners, and contributors. • Set up at least one collaboration channel (e.g. Gitter, Slack, StackOverflow tags if that is an option for you) • Create a communications document that sets out preferred communication channels and expected response times 📖
Nice To Have	• Define your project's values to help inform the Code of Conduct and to inform new members about your project's culture. • Create a recruitment plan for community development (beyond "just releasing and announcing") in preparation for the next stage.
Contributor Experience	
Must Do	• Contributor quickstart 📖 • Create issue and bug templates and issue tags
Nice To Have	• Provide good first issues that are easy for contributors to work on (e.g., requests for obvious test cases)
Stakeholder Management	

Must Do	• Alert your stakeholders that the project is released



On The Horizon

As you prepare to release and announce your project, you may want to explore:

- Best practices for building and stewarding your community
- Tools to manage user feedback and community contributions efficiently

For more things to consider, see the Key Questions in the Sapling stage

Lifecycle Stage: Sapling

About this stage

In this stage, you start building momentum! With more users and contributors, you may find yourself spending more time helping others, both inside and outside of your organization. At this stage stewardship of your project and community may be a full-time job for one or more employees.

To grow your community, you begin more outreach. Welcome new users and contributors with helpful documentation for both using and joining your project. Make sure community members understand how your project makes decisions by having clear explanations of your decision-making processes around both technical and community issues (including Code of Conduct violations). In addition, having a consistent brand and message will help you reach people both online and in-person (at meetups and conferences).

Key Questions to Consider

These questions are designed to help you prepare for this stage. Links to topical guides and examples are provided in the section to help you answer the questions.

- **Strategy:** Is my original open source strategy and plan on track? Does it need to be revised due to a shift in the business goal, the marketplace or the technology? Is the community forming aligned with the project goals?
- **Branding**: What resources can I use (external, internal, or community volunteers) to design a logo, build a media kit, and craft branding and messaging guidelines? Does my project name need trademark status in order to maintain quality across ecosystem partners (e.g. only projects that meet certain standards can use the trademark in relation to their project, or only approved events can use the project name officially)?
- **Production readiness**:
    - Documentation: Are there any use cases or groups of contributors whose concerns or interests are not represented in the current docs? Is there an easy way for people to contribute to the docs (through PRs or a wiki)? Do I see the same questions being asked over and over again in issues or other channels? Are the docs welcoming and useful to people with less experience in the domain?
    - Can I offer a bug bounty? What usability research still needs to be done? What accessibility feedback do I need to integrate?
- **Governance:** How is the direction of the project determined? If my project has been growing under a “Benevolent Dictator For Life (BDFL)” model of technical and community leadership, should I consider a transition to a technical steering committee or technical oversight board? If I have technical steering or oversight committees, is there a clear process through which users or contributors can become part of that group? Do I have Special Interest Groups (SIGs) or working groups to manage relevant aspects of my project, and a clear process for joining those groups? Is there a process for enforcing the Code of Conduct (CoC) and dealing with CoC violations?
- **Community Development:** Is my community growing as expected? Is there sufficient engagement in the community of both users and contributors? Are additional resources needed to support and manage communications channels, including social media and events? Can Developer Advocates support my project and provide outreach and advocacy? What is the overall investment needed to support the outreach and advocacy work for the project?
- **Contribution experience**: What tools are needed to deal with both the current load of issues, pull requests, or other contributor channels and expected future load? Do I have good first issues marked and sufficient documentation? Is the community welcoming of new contributors?
- **Stakeholder:** What do my stakeholders need to know as my project grows?

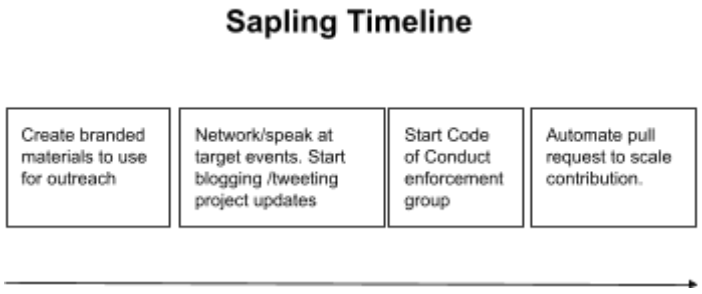
Key Steps To Take

In this stage, be sure to complete the Must Do steps. For a recommendation of priority order, see the Timeline below.

Focus Areas	Steps to Take
Strategy & Planning	

Must Do	• Check progress towards the project's goals and metrics. • Check the community development plan. Has the community changed? Does the community target plan need to be updated? • Determine what resources are necessary to support your project's progress, especially around community development and contributor experience. Investing in an Open Source Strategist and/or team is likely required to support the work in each pillar.
Branding	
Must Do	• Create a logo • Produce stickers to hand out at events • Create a presentation template
Nice To Have	• Develop brand and messaging guidelines for internal and external advocates to use • Create a media kit to help the press and community members find your logo easily (and get guidance on proper use) • Start the trademark registration process for your name, if you have determined a trademark would be helpful in achieving your project goals
Overachiever	• Design a mascot icon (e.g. Go Gopher) if you need additional brand recognition
Governance	
Must Do	• Publish (and follow!) process documents around technical governance • Consider establishing working groups or SIGs to handle specific questions or areas of interest • Establish and train a committee to enforce Code of Conduct violations.
Production Readiness	
Must Do	• Create tutorials for the top X use cases, especially enterprise use cases • Create and maintain a changelog • Perform usability research and create a usability plan • Security scanning • Accessibility work in-progress • Institute a public bug bounty program
Nice To Have	• Version lifecycle support policy 📖 • Skew testing • Specialized documentation for specific integrations or platforms 📖
Release Management	
Must Do	• Publish a public roadmap • Have documented and updated answers to the release management

	questions considered in the Sprout stage • Have a regular release cadence
Nice To Have	• If applicable, as number of external contributors grows, hold regular release meetings and retrospectives
Community Development	
Must Do	*Grow community:* • Attend target meetups and community events and/or secure Developer Advocates to help with outreach and advocacy. Have speakers use the branded presentation template. • Sponsor target community events • Add a blog to your project site and create a content calendar for your blog and social media (e.g. Twitter). Good things to feature include events, release information, roadmap changes, new partners, interviews with contributors, and information about related projects. *Accelerate user adoption and deepen usage:* • Provide a list of community events, or create your own events/meetups • Create user videos for your YouTube channel • Create case studies and demos; if appropriate, create a sample/toy application or a boilerplate repo • Solicit & publish testimonials from current users *Engage community:* • Create a rotation of internal contributors or a SIG to answer questions that arise in community forums (Slack, Stack Overflow, etc); set a SLO for time-to-answer for questions
Nice To Have	• Create an investment plan to support progressively more robust outreach and advocacy over multiple years • Foster community diversity (make sure governance committees have diverse representation, provide diversity scholarships, etc). • Sustainability programs: help support community leaders/members by providing leadership training, succession planning, community grants
Contributor Experience	
Must Do	• Train internal contributors in open source collaboration best practices • Scale contribution with “how to contribute” guides 📖 • Consider automated tools to support response to issues and pull requests (e.g. Git Probot)
Nice To Have	• Provide contributor mentoring • Set up contributor summits • Start contributor reward and recognition programs
Stakeholder Management	
Must Do	• Provide updates to stakeholders about progress and impact



On The Horizon

As your project grows momentum, you may want to explore next steps such as:

- Assess if your project is achieving the business goals or on track to do so
- Assess the staffing / resource needs to sustain a larger, more engaged community
- Assess the pressure points or risks arising due to the project's growth?

For more things to consider, see the Key Questions in the Mature stage

Lifecycle Stage: Mature

About this stage

In this phase, you are focusing on making your project sustainable. You have an active contributor base and a critical mass of users, and (if strategically important for your project) a significant partner ecosystem. Your project likely now has a lot of moving parts and a number of influential voices: in this stage most projects find a dedicated, full-time community steward a necessity in addition to full-time engineers.

Successful, sustainable projects depend on consistent policies and processes. You strengthen your governance by continuing to standardize your workflows and by creating committees and/or special interest groups to spread out the workload among other trusted and engaged community members.

Your community may spontaneously generate conferences and other larger events (such as hackdays or camps) or you may take the lead in organizing them. Community members advocate for the project by writing blogs, press releases, books, and by providing training both in-person and online.

At this point, you should have at a minimum, an Open Source Strategist to oversee implementation of activities within each pillar. Due to the level of effort in each pillar, this stage likely requires an investment in a team supporting the Open Source Strategist. This often includes someone focused on community communication and engagement.

Key Questions to Consider

These questions are designed to help you prepare for this stage. Links to topical guides and examples are provided in the section to help you answer the questions.

- **Production Readiness:** Are there any significant roadblocks to adoption remaining? Are there security concerns I need to address (e.g. CVE management)? Do I need to revise policies for managing deprecated sub-projects, packages, tooling, or for maintaining compatibility with ecosystem partners? Is my lifecycle support policy up to date?
- **Governance:** Does my project face any scaling issues or risk concerns? How can they be addressed by evolving policies, growing responsible committees, etc.?
- **Branding:** Do I have strong, recallable branding and branding elements for my project? Is there a consistent use of the project's name and logo?
- **Community Development:**
    - Growth: Is my community growing in the ways I intended? Are there any groups of users or contributors I didn't plan for, but should extend support to? Are new members joining the community at a slower or faster rate?
    - Education: How frictionless is the learning journey for my project? Are there gaps in my documentation that should be filled?
    - Engagement: Is the community at risk of burnout? Are there processes in place to allow active members to take breaks or reduce their workload without the project being impacted? Is the community welcoming? Is the community diverse?
    - Staff: Is it time to hire staff to manage community communication and engagement (e.g. event planning, blogs/social, Slack moderation)?
- **Contributor Experience:** Does my project provide welcoming onramps to new contributors? Am I still offering 'low-hanging fruit' for new contributors to work on? If a highly-motivated, highly-skilled contributor joined my project tomorrow, would they know where to begin? Do contributors feel good about the investment of time and energy they've made in my project?
- **Stakeholder Management:** Do my stakeholders know the fruits of my strategic plan and investment?

Key Steps To Take

These questions are designed to help you prepare for this stage. Links to topical guides and examples are provided in the section to help you answer the questions.

Focus Areas	Questions To Consider
Strategy & Planning	
Must Do	• Check progress towards the project's goals and metrics • Determine what resources are necessary to support the project's progress, especially around production readiness, community development, and contributor experience
Governance	
Must Do	• Identify areas where leadership needs to scale and work to ensure that critical responsibilities are shared to avoid bottlenecks or low "bus factors" • Ensure that there is clarity around how decisions are made for the project, especially in critical areas • Update policies and create working groups/SIGs as necessary to spread the workload
Nice To Have	• Public meetings to review significant governance or policy changes • Public annual reports of the state of the project
Branding	
Must Do	• Develop brand and messaging guidelines for internal and external advocates to use • Create a media kit • If you have registered a trademark, make sure you follow up with any necessary filings and renewals
Nice to have	• Design a mascot icon (like the Go Gopher) if you need additional brand recognition (or just for fun) • Create sticker templates for community members to create their own stickers for events or to trade
Production Readiness	
Must Do	• Review the practices created in the Sapling stage and update any processes that have become awkward or creaky • Provide sample code or tutorials for multiple languages/clients and integration points 📖 • Consider creating a stand-alone documentation site 📖 • Your CI/CD, testing, and security scan processes should be automatic and well-understood • Create version support and support lifecycle policies • Create documentation for upgrading or downgrading versions 📖 • Create a policy for vulnerability disclosures • Dependency regression testing • Project is accessible • Usability testing/research is done and the results are integrated • Encourage ongoing user feedback and have a process for receiving

	and acting on suggestions
Nice To Have	● Create interactive API documentation (where appropriate) ● Provide or support community-driven workshops or third-party classes ● If applicable, consider beginning a certification program
Release Management	
Must Do	● Hold regular release meetings and retrospectives ● Have a regularly updated public roadmap ● Review your release practices and update any that have become inefficient or time-consuming ● Review any recent release retrospectives and prioritize any changes needed in your process
Nice to Have	● Consider using cryptographically-signed binaries, if applicable
Community Development	
Must Do	*Grow Community:* ● Review your community plan. Are you on track? ● Strive to understand how people find, explore, and join the community. Identify blockers and solutions to remove them. ● Conduct more formal PR programs to highlight project momentum, case studies, and updates. ● Sponsor and speak at key community conferences and meetups. *Educate Community:* ● Create an ambassador program to help key community members advocate for the project. ● Secure Developer Advocates to support outreach. ● Conduct UX study of user learning paths and strengthen online learning materials to accelerate the user's adoption journey. ● As an experiment, do a scan of your busiest communications channel for a day and see if every reasonable question can be linked to documentation that would answer it *Engage & Support Community:* ● Foster community diversity (make sure governance committees have diverse representation, provide diversity scholarships, etc). ● Sustainability programs: help support community leaders/members by providing leadership training, succession planning, community grants ● Hire a dedicated person to manage/write for blogs, social media, and engage with community members in your communication channels
Contributor Experience	
	● Provide contributor mentoring ● Set up contributor summits ● Create contributor reward and recognition programs

Stakeholder Management	
	• Give updates on current progress and show how investments are delivering outcomes on both the business plan and the open source project. You may find that you need to secure long term investments to support the continued success of your program.

Lifecycle Stage: Snag / Deprecated

About this stage

All good things must come to an end, and you may find that your project no longer has an active community of users or contributors and should be deprecated. The goal of this stage is to gracefully turn off the lights and guide your community to new resources.

Key Questions to Consider

- Is there anyone who can take over the project?
- Are there similar projects that a user could use instead?
- What are the best ways to tell users about the end of life decision?
- How much notice should users receive before end of life?
- What security updates, if any, should be offered after end of life?
- What should be done with the project page and community assets once a project is deprecated?

Key Steps To Take

Focus Areas	Steps to Take
Strategy & Planning	
Must Do	• Create a list of affected groups and partners for consultation and messaging • Determine your deprecation timeline • Create plans for communicating and documenting project deprecation
Governance	
Must Do	• Determine the applicable policies and groups that must be consulted and followed in your deprecation process • Perform all required consultations with the community.
Production Readiness	
Must Do	• Determine a long term support plan and who provides it

	• Update documentation to inform users that the project has been deprecated • Provide links to similar projects for reference 📖
Release Management	
Must Do	• If applicable, revise release documentation to indicate that only fixes needed under the long-term support plan are being released • Follow the current best practices for deprecating repos at any hosting site (e.g. GitHub, NPM)
Community Development	
Must Do	• Create and implement a communication plan • Ensure a reasonable amount of time between communication and end of life
Contributor Experience	
Nice to Have	• As part of your communication plan, ensure all community members are alerted to the end-of-life decision by making announcements in multiple channels, sending a special email announcement, etc. • Recruit community members to help create lists of possible replacement projects (they may have moved on to these projects and would welcome the chance to share them) • Consider asking community members (especially those who have moved on from the community) to share their favorite memory of working with or contributing to your project. Share on social media or as a final valedictory blog post.
Stakeholder Management	
Must Do	• Advise the stakeholders of the decision to deprecate and gain any necessary approvals

Appendix

Using open source to achieve business goals

The decision to release software as open source should be taken strategically, in order to achieve a specific business goal. (For more information on open source strategies, consult "Why Open Source?")

If you want to	Then, consider	Description

achieve this business goal	**these Open Source strategies**	
Create business efficiencies through knowledge and resource sharing	Share your work	Assuming there is no risk of jeopardizing future monetization opportunities, one can open source to create efficiencies like tools that help groups stop reinventing the wheel or to foster knowledge sharing by passing on journals or data sets. Additional benefit: no community investment is needed since communities don't form around the project.
Accelerate adoption of products and services Share or reduce development costs for select functionality	Establish or support an open standard	A widely-adopted open standard provides a large ecosystem to leverage for your product or service that is built on that standard. An open standard can create a large contributor base that accelerates innovation, benefiting your offering.
Break into a new market that has established players	Disintermediation	This strategy creates an opening to compete in a new field by eliminating vendor lock out and creating portability. For example, by open sourcing Kubernetes and by providing a useful tool that happens to be an abstraction layer, Google made it easier for people to use multiple cloud service providers, or switch between them.
Accelerate adoption of platform offering	Platform & ecosystem	Open sourcing proliferates the extensibility mechanisms inherent in a platform offering. This open source strategy attracts contributors who build value on top of the platform. Plus, it attracts an ecosystem of vendors, service providers, ISVs and users who are invested in the platform's success because they built a business around it.
Make existing technology more relevant, more valued in the marketplace	Change focus	This open source strategy is for projects that can enhance or re-invigorate existing technology. By open sourcing, the enhancements can be shared throughout the market segment, leveling up all offerings.

Open Source Lifecycle

Once you identify the open source strategy you will use to achieve a business goal (and have stakeholder support), you will start your open source project, which will go through a lifecycle

from inception to maturity or deprecation. Some projects will reach maturity quickly with little investment, while other project goals will take longer and require more investment to achieve.

Open source projects have a lifecycle: they are born, they grow, they mature, and (sadly) one day they are no longer used or useful, and they die. In the model below, inspired by the lifecycle of trees, typical signals of each stage are outlined. Not all projects progress through the lifecycle at the same pace or in the same way, so use this as a guide, rather than a template.

	Define	**Signals that you are in this stage**	**OSS Strategy Achieved by...**
SEED	“Perhaps I should open source?”	You think that there could be benefit in open sourcing. You determine if open sourcing supports a business goal. If yes, you gain stakeholder approval.	----
SPROUT	“Just Launched”. Code is released in compliance with org guidelines.	You release code in compliance with your organization’s licensing standards. You may start a GitHub project. You release very basic documentation. You announce via your company blog or social media channels	“Share Your Work”
SAPLING	“Building Momentum”. A small community forms.	You find yourself spending more time helping others. You work with them on questions, code contribution, etc. You encourage enthusiastic participants to grow into leadership roles in the project and become stakeholders. To be more responsive you improve documentation, use Stack Overflow, or create a larger website. To grow, you begin more outreach. You may see meetups or other in-person gatherings around the project.	
MATURE	“Become Sustainable”. A larger, more engaged community forms	You find that you need policies and automation to manage contributors. You also need governance to expand project maintainership to include other stakeholders. An ecosystem has formed around your technology. People hold events to discuss the technology. Employees of your org and others are writing press releases, books, and trainings.	Open Standard, Platform adoption, Disintermediation
SNAG	Deprecated	No one is using or maintaining the open source project	

Open Source Stewardship Pillars

After you release your code, it is time to build and steward your community. As projects grow, they need support in areas outside of (and in addition to) the code itself. Project stewardship involves developing these pillars in response to a project’s needs at each stage.

Pillars	Description	Outcomes/Deliverables
Strategy & Planning	An understanding of the business goals and which open source strategy to use to achieve it and how to measure success.	Business goal - open source strategy - metrics
Governance	Policies that define the systems through which decisions are made and how the community works together to release software and to become or remain sustainable, healthy, and engaged	Values - Code of Conduct - Decision-making policies - Committees
Production Readiness	Clear processes for ensuring the project is committed to the security, reliability, and usability of the project code in production environments	User documentation - CI testing - Version support - Support lifecycle - User feedback - CVE management
Release Management	Clear policies, processes and actions that ensure timely, well coordinated releases	Roles defined - versioning policy - release process / policies - patch management - testing requirements -- branch management - release retrospectives
Branding	Sufficient name, brand identity, and brand messaging to enable others to find and refer to the project easily	Name - Trademark - Logo - Style guide - Reputation Management
Community Development	Design a plan to recruit users, ecosystem partners, and/or contributors into the community via outreach and advocacy. Support members on their adoption journey and foster a healthy engaged community that helps each other.	Events - PR - Marketing - Code of Conduct Enforcement - Communication Channel Management (social, website, blogs, StackOverflow, Slack)
Contributor Experience	Tools and resources for contributors (code, events, docs, etc) that help them learn how to contribute and continue to do so with a rewarding and sustainable experience.	CLA - Contributor Guide - PR lifecycle - Issue lifecycle
Stakeholder Management	Ensuring alignment and maintaining buy-in from stakeholders such as engineering and product management leadership, community members, and users.	open source Strategy stakeholder management - Community stakeholders