

# TECHNOLOGY

## Phase I: Laying the Groundwork



### Activity 1: Who Are Your Stakeholders?

#### Goals

1. Identify and prioritize the community's technology stakeholders
  - A stakeholder is any individual, group, or organization within or outside the project who is impacted by its outcome or who has an interest in its success
2. Create a high-level map of the technology stakeholder's characteristics
3. Identify the program's goals for high-priority stakeholder groups

#### Prerequisites

CE Activity: Who Is Your Community?

#### Who Should Participate?

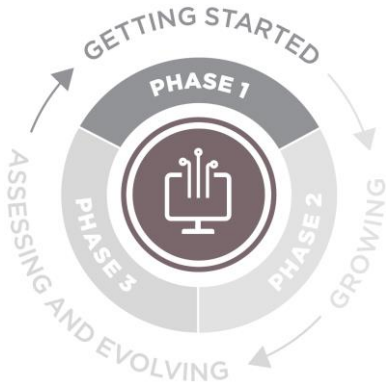
Program leadership (strategic thinkers), Program management (tactical thinkers), Program staff (operational experience)

#### Length

60-90 minutes

#### Activity Instructions

1. On a whiteboard or shared online document, identify your potential technology stakeholder groups. Using sticky notes (real or virtual) works best, as ideas will be moved around in later steps.
  - It's ok to be granular - your team will eliminate duplicates / combine stakeholders in the next step.
  - Stakeholder examples: end users, QA testers, code contributors, software engineers, sysadmins/IT staff at institutions using the software, program staff at related OSS programs, functional requirements contributors, etc.
2. Move the sticky notes around to categorize your stakeholders into groups.
  - Group examples: Technical contributors, Service providers, etc.
3. Select 3-5 groups to prioritize over the next year. An interest/influence grid or sticker vote may be helpful in prioritizing.
  - An interest/influence grid plots stakeholder groups against two axes: Interest and Power/Influence and then suggests a level of engagement based on their place on the grid. A sample grid is below.
  - In a sticker vote, each participant is assigned a number of stickers - these can be physical stickers in an in-person event or a specified piece of text (e.g. +1) in a virtual environment. Participants place their stickers or text alongside the options they're voting for, according to the parameters of the exercise.
4. For each prioritized stakeholder group in your diagram, discuss the following questions:
  - What are our goals for each user group or technology stakeholder group?
  - Are there shared or related goals across stakeholder groups? What are the opportunities and areas of collaboration? How can our community work together to create and achieve things?
5. If you have completed Technical Skills Inventory Part 1, you may also answer:
  - What technology skills are required to make our mission a reality? Do our prioritized stakeholder groups have these skills?

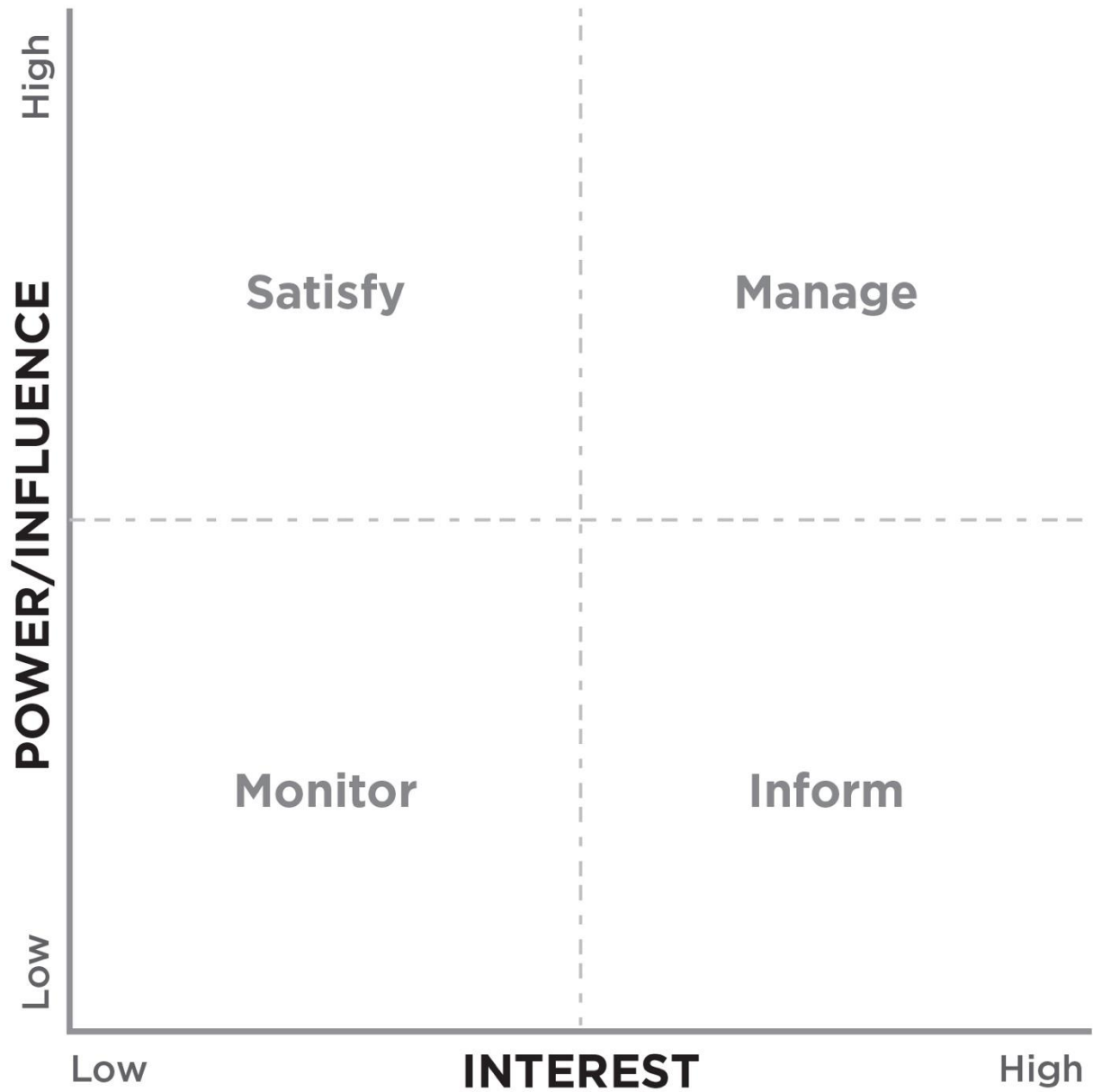


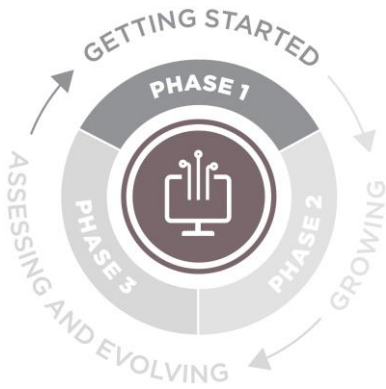
# TECHNOLOGY

Phase I: Laying the Groundwork



## Activity 1: Who Are Your Stakeholders?





# TECHNOLOGY

## Phase I: Laying the Groundwork



### Activity 2: Technical Skills Inventory, (Part One)

#### Goals

1. Create an inventory of what technical skills are needed for technical staff, contributors, and users to develop, support, and maintain the platform.

#### Prerequisites

None

#### Who Should Participate?

Program management (tactical thinkers), Program staff (operational)

#### Length

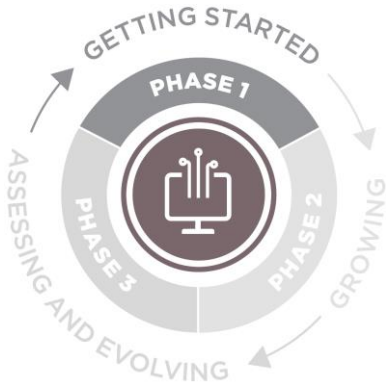
This activity does not need to be done as a group; it can be completed asynchronously/ collaboratively in a shared document.

#### Activity Instructions

1. Create a list of the technical roles held by program staff/stakeholders. Roles should be fairly granular, describing one aspect of developing, maintaining, or supporting your application.
2. Fill out the below template to add the technical skills required to fulfill the roles you listed. Examples in blue.
3. Once it's complete, sharing the skills inventory as part of your public documentation is a great way to let potential users and contributors know what types of skills are needed to take on certain tasks.

#### Once complete, this activity can be used in several additional contexts, including:

- Roadmap planning: ensuring that major deliverables on the roadmap are not all clustered around a certain skill set (and therefore a certain person).
- Community building: If users self-hosting is important to your program, the skills inventory can be used to compare skills required to install, upgrade, and maintain the software against skills that your end users have or have access to in their organizations.



# TECHNOLOGY

## Phase I: Laying the Groundwork



### Activity 2: Technical Skills Inventory, (Part Two)

**Example roles (depending on tech stack, may need to qualify with frontend, backend, etc.)**

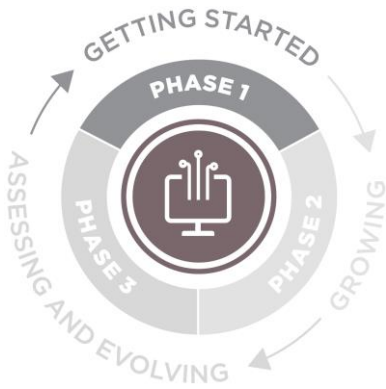
Technical lead, Code contributor, System administrator (e.g. install, upgrade), End user, Migration specialist, Technical support for End user

**Example skills (include specific tools in table, e.g. HTML for Web design or development)**

Web design, Web development, Assistive technology, Database, Data warehousing, Data analysis, GIS, Platform/OS, Quality assurance/Testing, Reporting, Security, Server

**Sample Technical Skills Inventory – Examples in *Italics***

Role	Required skill	Skill Level		
		Novice	Intermediate	Advanced
<i>Developer: Front end</i>	<i>Angular/Typescript</i>	X		
	<i>HTML</i>		X	
	<i>Bootstrap</i>	X		
	<i>SAAS/CSS</i>		X	
<i>System administrator (install, upgrade)</i>	<i>Command line</i>		X	
	<i>Package manager</i>		X	
<i>Report writer</i>	<i>SQL</i>			X
	<i>Crystal Reports</i>			X
...	...			



# TECHNOLOGY

## Phase I: Laying the Groundwork



### Activity 3: Personas and Pathways

#### Goals

1. Learn how to bring contributors onto your project by using tools called "personas" and "pathways"
2. Help your program plan and test how you'll interact with new contributors
3. Imagine what is really involved for new contributors to succeed

#### Prerequisites

Suggested: Who Are Your Tech Stakeholders and Technical Skills Inventory, Part 1.

#### Who Should Participate?

Program management (tactical thinkers), Program staff (operational expertise)

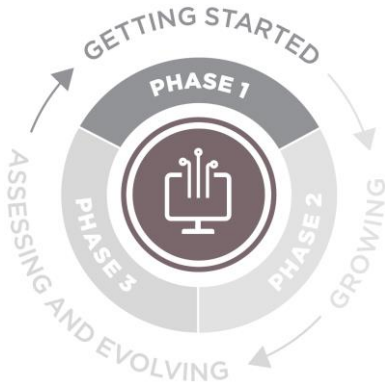
#### Length

60-90 minutes

#### Activity Instructions

Complete the Mozilla Open Leadership Activity: Contributor Personas and Pathways:

<https://mozilla.github.io/open-leadership-training-series/articles/building-communities-of-contributors/bring-on-contributors-using-personas-and-pathways/>



# TECHNOLOGY

## Phase I: Laying the Groundwork



### Activity 4: Landscape Analysis

#### Goals

1. Understand where your program fits in the tech landscape
2. Use results to innovate, make decisions, identify opportunities for collaboration, increase usefulness/effectiveness

#### Prerequisites

Gov Activity: [Mission/Vision](#)

Activity: [Who Are Your Technology Stakeholders](#)

#### Who Should Participate?

Program management (tactical) with initial brainstorming input from program leadership and program staff.

#### Length

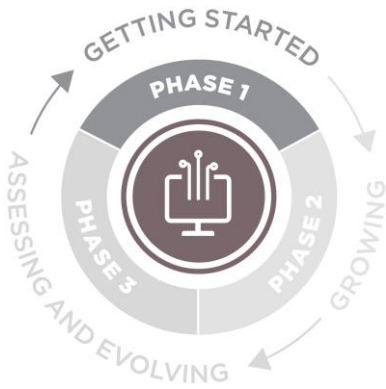
90 minutes

#### ACTIVITY INSTRUCTIONS

1. Build as comprehensive a list as possible of the competitive landscape - programs that are engaged in roughly the same work, serving roughly the same stakeholder groups. This list can be created during an in-person brainstorming session, or offline through a shared document or virtual whiteboard.
2. Create an assessment template that includes (at least) the following categories:
  - a. Program name
  - b. High level purpose
  - c. Core features/functionality
  - d. Target audiences
  - e. Mission, values and vision — What are the specific, tangible goals they're trying to accomplish?
  - f. Unique value proposition — What does the program claim to do that is different from other programs?
  - g. OSS license (or note if it's a proprietary application)
3. Analyze the landscape by thinking through the following questions:
  - a. How is your program different? How is it the same?
  - b. What are you doing better? What can you highlight in your messaging and communications that is unique and will resonate with your target stakeholders?
  - c. Where are your competitors excelling? What can you learn from them?
  - d. Where are your competitors falling short? Are there any strategies that aren't working? Could you do them differently or better?
  - e. What "gaps" do you see? Are there missed angles or opportunities? Could you fill those gaps? How does this information inform your own point of view and messaging?

The results of your analysis may be useful as noted in the above goals, and as groundwork for Technology Facet activities around roadmap development.

LYRASIS holds the copyright to this Activity and provides it for free use, sharing, copying, distribution and adaptation with attribution via the Creative Commons Attribution 4.0 International License (CC BY 4.0) available at: <https://creativecommons.org/licenses/by/4.0/>; Activity adapted from: Alameda, Leeann. "The Power of Competitor Analysis for Nonprofits." NonProfit PRO. Accessed February 18, 2021. <https://www.nonprofitpro.com/article/the-power-of-competitor-analysis-for-nonprofits-and-how-to-get-started/>.



# TECHNOLOGY

## Phase I: Laying the Groundwork



### Activity 5: Buy a Feature Game

#### Goals

1. Help users prioritize among a list of pre-defined (but not yet developed) features
2. Promote cooperation among end users

#### Prerequisites

Example: None

#### Who Should Participate?

Program management (tactical thinkers), Program staff (operational experience), End users

#### Length

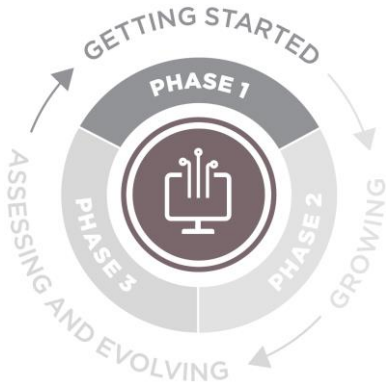
60 minutes

#### Activity Overview

This is a game to get end users to prioritize among a list of pre-defined (but not yet existing) features. Users get a limited budget of play money to fund their preferred features (a sample game board is linked in the instructions below). The instructions are divided into two sections, the first that a subset of program management representatives should complete, and the second with a group of end users.

#### Program Team Activity Instructions - Complete First

1. Program management representatives (e.g. PM, Tech Lead) work together to create a list of feature requests or potential roadmap additions for end users to help prioritize.
2. Define a budget for your participants, e.g. \$100 per person or organization.
3. If holding the activity virtually, create a spreadsheet to track the contributions. Suggested setup:
  - a. In the first worksheet, create a list of all features you'd like participants to spend money on.
  - b. In subsequent worksheets, copy that list of features, and add a column for participants to note how much of their budget they would spend. It's helpful to add a sum at the bottom so they can see how much they've spent.
  - c. Back in the first worksheet, add a column that rolls up the dollars spent by participants in each individual worksheet to see which features were most highly prioritized.
  - d. A sample worksheet with pre-set calculations is available [here](#).
4. If holding the activity in person, write the names of the features on cards, and put a small container in front of each card. Provide participants with a stack of play money and have them contribute to each feature.
5. Schedule the activity and invite guests.



# TECHNOLOGY

## Phase I: Laying the Groundwork



### Activity 5: Buy a Feature Game

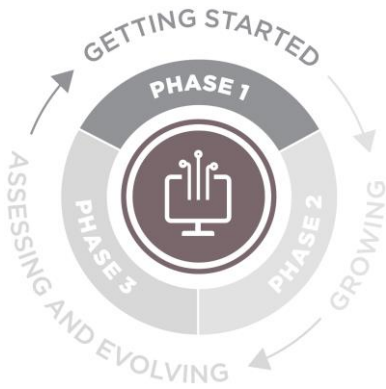
#### End User Instructions

1. **Have the participants buy features.** Invite participants to allocate their budget among the features they want.
  - a. Optional: Allow users to suggest new features that are not on the list.
  - b. Note: This can be done synchronously as a group or asynchronously in which your invitees are given a set time (e.g.) a week in which to purchase their features.
2. **Analyze outcomes.** Analyze which features get the most attention, which are funded, and which ones are not.

#### Output

A prioritized list of not-yet-developed features to assist with future roadmap development.





# TECHNOLOGY

## Phase I: Laying the Groundwork



### Activity 6: Community QA Testing

#### Goals

1. Review program's current QA practices and evaluate whether they are structured in a manner that can be supported by the community
2. Identify places for improvement
3. Conduct a successful round of Community QA

#### Prerequisites

None, but [Activity: Recognition and Contributions](#) may be useful.

#### Who Should Participate?

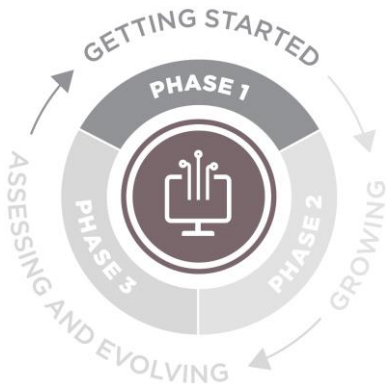
Program management  
(tactical thinkers)

#### Length

X minutes

#### Activity Instructions

1. Fill out the checklist on pages 2-3. Rate whether your program has these elements in place, does not, or they're in progress. You can also note if you are unsure, or if the metric does not apply.
2. Work together to identify 3-5 elements in the checklist that could be improved before your next release. Plotting elements on an impact/effort matrix (example on page 4) can aid prioritization.
3. Identify who on the program team will be accountable for the element - either taking on the responsibility for creating/improving the element or assigning the task to someone else and following up on its completion.



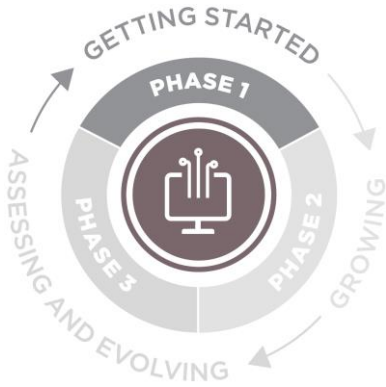
# TECHNOLOGY

## Phase I: Laying the Groundwork



### Activity 6: Community QA Testing

Testing tools	Yes	No	In progress	Unsure	N/A
Clear locations for QA communication / documentation: QA docs for program staff QA tests for end users Bug tracker instructions Bug tracker Q&A / Help Other _____					
Instructions for which platform to use for which purpose <i>Jira for bugs, Slack for questions, wiki for test plans, etc.</i>					
Volunteers	Yes	No	In progress	Unsure	N/A
Cadre of dedicated testers <i>Maintain, review list on regular basis</i>					
Orientation for new testers					
Clear instructions for new and returning testers <i>How to choose a test, file a bug, etc.</i>					
“Good first test” flag for new testers					
Method for ID’ing community members who could take on additional responsibility					
Recruitment plan for new testing volunteers					
Events	Yes	No	In progress	Unsure	N/A
Schedule for QA testing process					
Themed and time-bound QA events <i>New releases, beta features, usability, etc.</i>					
Events adapted for different times zones and/or languages					
Event promotion across multiple platforms					
Specific and clearly communicated goals for the event					
Reward / recognition plan					



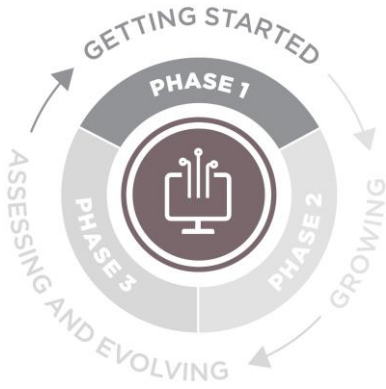
# TECHNOLOGY

## Phase I: Laying the Groundwork



# Activity 6: Community QA Testing

Documentation	Yes	No	In progress	Unsure	N/A
Roles and responsibilities <i>Who manages the events – staff, community members? What are their roles?</i>					
What is QA / QA testing <i>Non-jargon essential</i>					
How to create test plans					
How to QA test plans (really) <i>Are the steps outlined in the test plan really those taken by the end users?</i>					
How to organize / hold a testing event					

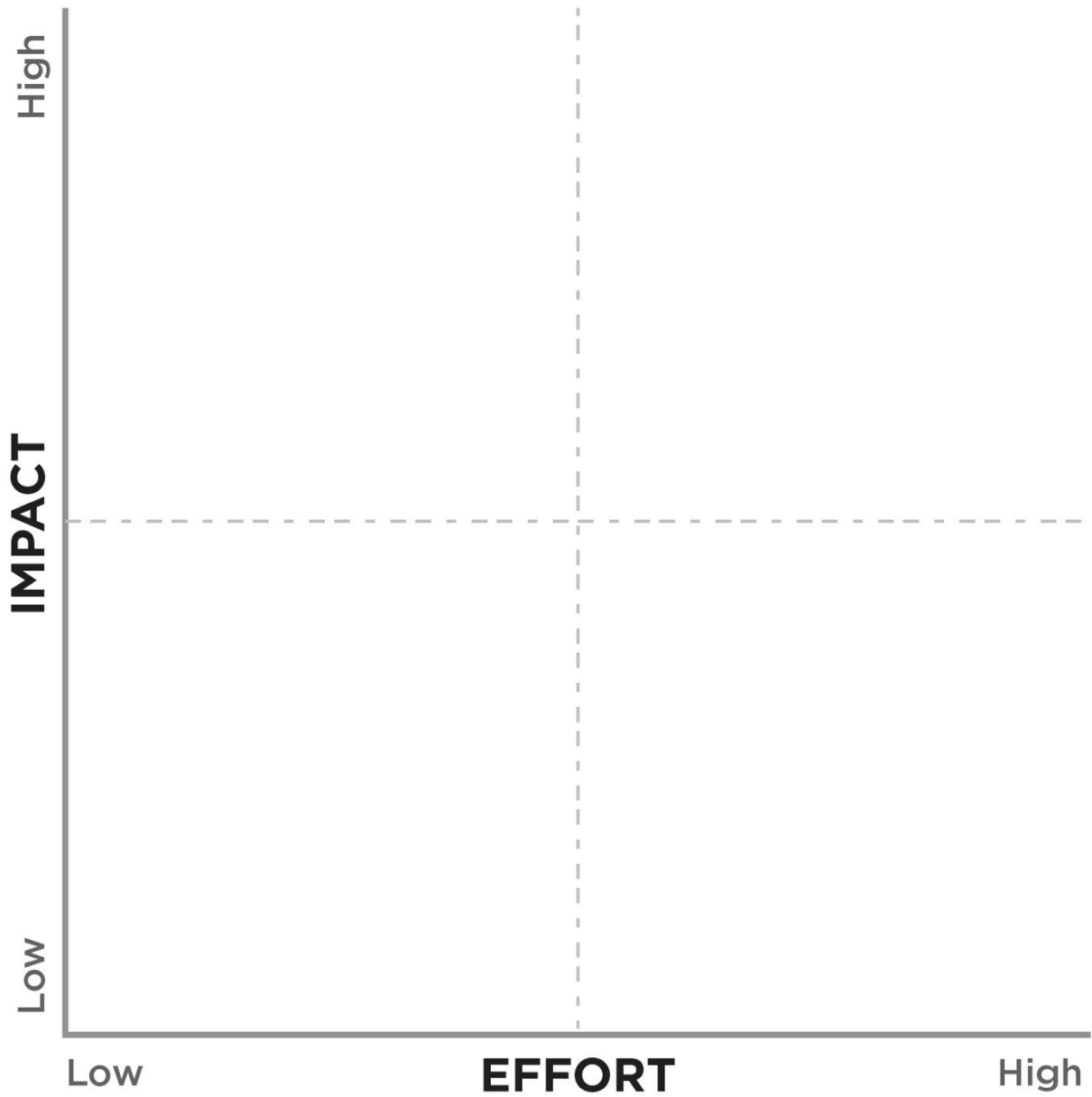


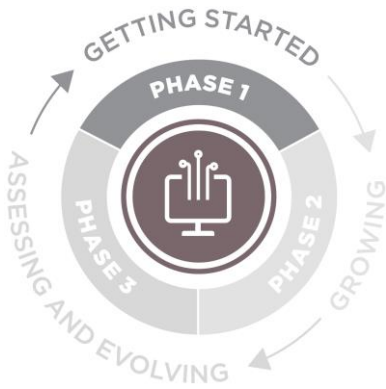
# TECHNOLOGY

Phase I: Laying the Groundwork



## Activity 6: Community QA Testing





# TECHNOLOGY

## Phase I: Laying the Groundwork



### Activity 7: Docs Friction Logging

#### Goals

1. Understand how usable your current documentation is via friction logging with documentation users.

#### Prerequisites

None

#### Who Should Participate?

Documentation creators and consumers

#### Length

30-60 minutes for documentation consumer, additional time for creators to review results and update documentation as needed.

#### Definition

A friction log is a document that lists the elements that make a tool hard to use but frames it around a narrative or customer use case. The critical difference between a friction log and a bug list is that the friction log puts the issues in context. It tells the story of the entire user experience from start to finish.

#### Pre-Work

Decide which elements of your current end-user documentation you'd like to evaluate. You may wish to evaluate "user manual" type documentation for your application (e.g. how to create an account or how to upload an image), or program-focused documentation (e.g. how to contribute code or how to submit a bug report). The most useful friction logs will evaluate documentation that lots of folks are likely to use.

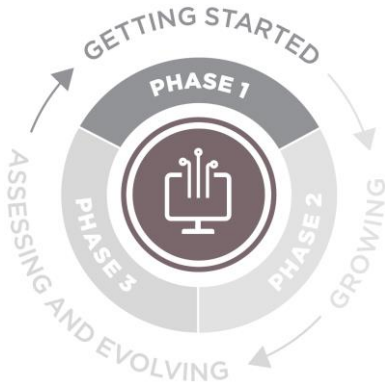
A good example of a friction log can be found about halfway down this blog post:

<https://www.trychameleon.com/blog/friction-logs>

#### Instructions for Program Staff

1. Determine which elements of your current end-user documentation you'd like to evaluate.
2. Identify community members to serve as loggers. Make choices based on what you're testing; and try not to choose people who are so familiar with the system that they'd know how to skip over the hard or confusing parts.
3. Create a friction log template (example on page 4 below). Include at least the following questions:
  - a. Logger's name
  - b. Platform/language/browser (if relevant)
  - c. Date
  - d. Product used for the log (if your program supports >1)

LYRASIS holds the copyright to this Activity and provides it for free use, sharing, copying, distribution and adaptation with attribution via the Creative Commons Attribution 4.0 International License (CC BY 4.0) available at: <https://creativecommons.org/licenses/by/4.0>. Friction logging definition and exercise adapted from: Hammerly, Aja. "An Introduction to Friction Logging." DevRel.net, July 8, 2018. <https://devrel.net/developer-experience/an-introduction-to-friction-logging>.



# TECHNOLOGY

## Phase I: Laying the Groundwork



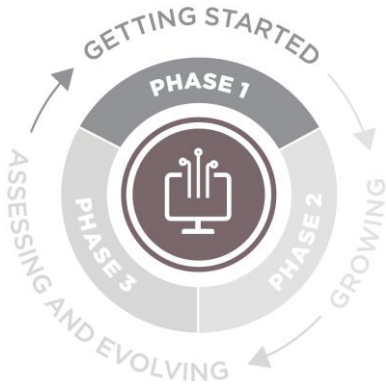
### Activity 7: Docs Friction Logging

- e. Brief description - no more than two short sentences - of the scenario/use case the logger is following the docs for - such as:
  - i. I want to figure out how to contribute code to the platform.
  - ii. I am trying to add a new user and give them account permissions.
4. Include a link to the documentation (or attachment if it's not available online)
5. Provide the friction logger with the logging template
6. Once the logger has completed the work, evaluate the comments, and determine if any changes or improvements need to be made to the documentation

#### Instructions for Friction Logger

1. Using the provided template and following the instructions in the documentation, create a log of what you did to complete your scenario / use case, and any reactions you had while doing so. For example:
  - a. If you searched a wiki or GitHub repository, write down the search terms
  - b. If you typed commands into a command line, copy and paste your command into the log
  - c. Make note of any places where the documentation was not clear, or the instructions didn't work as written
2. Record your reactions for each step, such as "Now I'm frustrated," or "Copied and pasted this from the docs, didn't bother reading the prose."
3. Stoplight-colored highlights (red, yellow, green) in the log can be used to point out particularly excellent or problematic parts of the experience.
  - a. Green: Something delightful. Maybe there was a great example, or something worked right out of the box without requiring extra configuration.
  - b. Yellow: Something that was frustrating. Perhaps you couldn't find relevant docs after 10 minutes of searching.
  - c. Red: Places where you would have given up if this wasn't your job.

LYRASIS holds the copyright to this Activity and provides it for free use, sharing, copying, distribution and adaptation with attribution via the Creative Commons Attribution 4.0 International License (CC BY 4.0) available at: <https://creativecommons.org/licenses/by/4.0/>  
 Friction logging definition and exercise adapted from: Hammerly, Aja. "An Introduction to Friction Logging." DevRel.net, July 8, 2018. <https://devrel.net/developer-experience/an-introduction-to-friction-logging>



# TECHNOLOGY

## Phase I: Laying the Groundwork



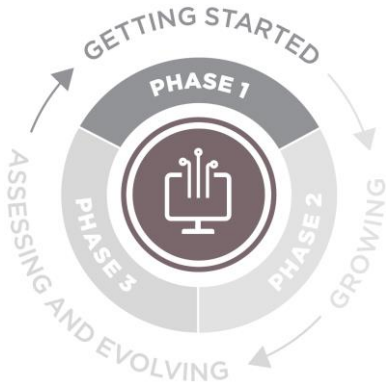
### Activity 7: Docs Friction Logging

#### Example

Scenario: Create a Microsoft Teams account

Steps:

1. Use Google to look up "Microsoft Teams"
2. Click on the first result - "Video Conferencing, Meetings, Calling | Microsoft Teams"
3. "Sign up for free" is the first thing I notice on the screen, so I click on that button
4. I enter my email and click Next
5. I am using this to participate in volunteering, so I select "For work and organizations"
6. Enter password? They want the password to my email account, or do they want me to create password for this Teams account? This is confusing.



# TECHNOLOGY

## Phase I: Laying the Groundwork



### Activity 7: Docs Friction Logging

#### Friction Log Template

Name:

Platform/Language/Browser:

Date:

Platform:

Scenario / Use Case (two short sentences):

*Create a narrative that tells the entire story, from start to finish, of your scenario or user story. Include every step you took and reaction you had. Stoplight-colored highlights (red, yellow, green) or prefixes in the log can be used to point out particularly excellent or problematic parts of the experience: **Green**: Something delightful, **Yellow**: Something frustrating, **Red**: Places where you would have given up if this wasn't your job.*