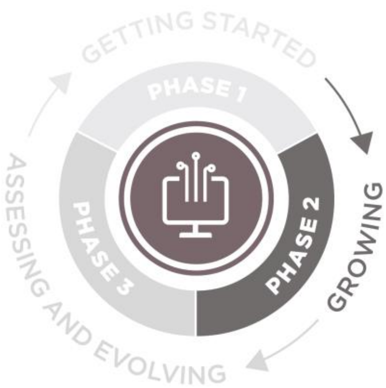


TECHNOLOGY

Phase II: Expanding and Integrating

Activity 8: Shadow Observations

Goals

1. Gain insight about users' jobs, pains, and gains
2. Learn about users' day-to-day work in a real-life setting
3. Incorporate end-users experience into program planning

Prerequisites

None

Who Should Participate?

Program management (tactical thinkers)

Length

60 minutes per shadow

Activity Instructions

In shadow observations, one or more members of the program staff observe an end user interacting with the software. Generally, the people observing the user don't interfere with the user's actions - the idea is to see how they go about their daily work as normal.

This activity can be done in-person, or online via screen-sharing.

Pre-Work

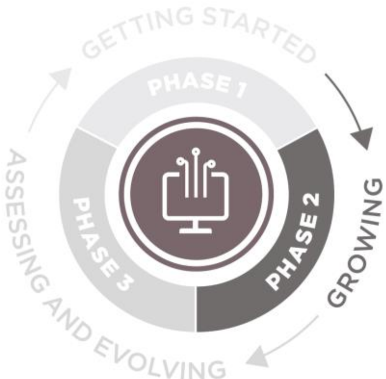
Work with program colleagues to:

- Determine the goal of the shadowing exercise - do you want to see how a new feature is being used, get information to help with prioritization, etc.?
- Invite the community to volunteer for shadowing. Your invitation should include:
  - Brief description of what shadow observation is
  - Length of observation
  - Platform
  - Whether it will be recorded (recommended if online)
  - What specific interactions you'd like to see, e.g. logging in, adding content, running a report, etc.

During the Observation

- Beyond the initial prompt (show me how you…), try not to interfere with the user, just let them user the software normally
- Ask the user to narrate their actions as they complete them
- Keep track of your observations as they happen, the template below may be helpful
- Hold questions until the end when the shadowing is complete

Materials adapted from: "Shadowing in User Research - Do You See What They See?" The Interaction Design Foundation. Accessed December 15, 2020. https://www.interaction-design.org/literature/article/shadowing-in-user-research-do-you-see-what-they-see.



Activity 8: Shadow Observations

After the Observation (Still with User)

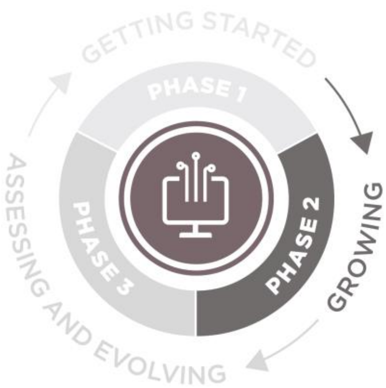
- Ask questions! Ask the user to clarify anything you didn't understand or would like more detail about.
    - Consider using a technique such as the Five Whys, which asks "why" questions in response to five consecutive answers. This prompts the user to examine and express the underlying reasons for their actions.

After the Observation (No Longer with User)

- What did you learn?
- What positive interactions did you observe?
- What pain points did you observe?
- How can you fold your findings into future road mapping and prioritization exercises?
- Summarize the high-level results and share them with the community
- Work to prioritize concerns/needs and fold into road mapping activities

Sample Notes Template

Time	Activity (What I See)	Notes (What I Think)



TECHNOLOGY

Phase II: Expanding and Integrating

Activity 9: Recognition & Contributions

Goals

1. Understand how the program currently recognizes contributions
2. Understand the kind of recognition that is most appreciated/motivating to each kind of contributor
3. Discover how the program can improve in this area

Prerequisites

Tech Activity: Who are Your Technology Stakeholders?

Who Should Participate?

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

Length

60-120 minutes

Activity Instructions

As a group:

1. Discuss the 6Rs (listed below).
2. Which of the elements are currently incorporated in your program?
3. Which are not currently incorporated?

Divide into small groups of 4-5 people. In each group:

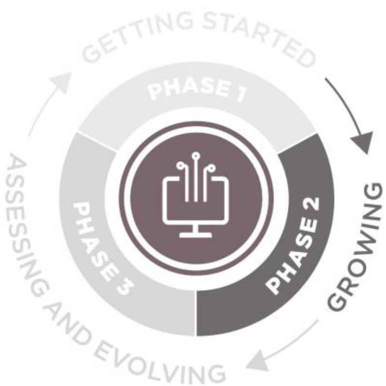
1. Select a specific stakeholder group from Tech Activity 1 (linked above - it's ok for the same stakeholder groups to be discussed by more than one small group)
2. Take 15 minutes to talk through what types of recognition might be important for the chosen stakeholder group
3. Select a representative to report back to the larger group

As a group:

1. Hear reports back from small groups
2. Discuss themes and gaps that need to be addressed to keep current contributors engaged and to entice new contributors
3. Prioritize and create an action plan to incorporate new forms of recognition
4. Who will be accountable for ensuring the plan is created / carried out?
5. Who will be responsible for doing the work?

For the future / Potential next steps: The information gathered during this exercise can inform working groups or specific outreach activities. Consider if there need to be higher level program changes to accommodate the recognition needs (i.e. more representation on governance groups or different voting rights).

Activity adapted from "8. Increasing Participation and Membership | Community Tool Box."



TECHNOLOGY

Phase II: Expanding and Integrating

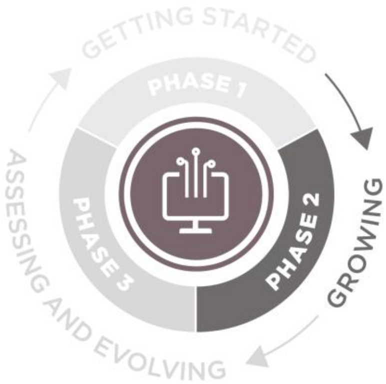
Activity 9: Recognition & Contributions

The 6Rs

Contributions are critical for community and open-source software. Understanding what motivates potential and current contributors is essential.

There are 6 crucial recognition qualities. These are not “one size fits all” but these can be a helpful way to consider what is important to different contributors. They include:

1. **Recognition**: People want to be recognized for their contributions.
2. **Respect**: People want their values, culture, ideas, and time to be respected and considered in the organization's activities.
3. **Role**: People want a clearly meaningful role in the coalition that makes them feel valuable and in which they can make a contribution.
4. **Relationships**: People want the opportunity to establish and build networks both professionally and personally for greater influence and enjoyment.
5. **Reward**: People expect the rewards of participating in a collaborative partnership to outweigh the costs and to benefit from the relationships established.
6. **Results**: People respond to visible results that are clearly linked to outcomes that are important to them and that they can clearly link to their participation in the coalition.



TECHNOLOGY

Phase II: Expanding and Integrating

Activity 10: Value Propositions for Job Descriptions

Goals

1. Create a value proposition for a new program position (e.g. community manager, technical lead) to help articulate significance of allocating resources to program leadership
2. Optional: Create a job description for a new program position

Prerequisites

Job description (optional). If you don't have a job description yet, having samples on hand of job descriptions of similar roles at other projects can be helpful as a starting point.

Who Should Participate?

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

Length

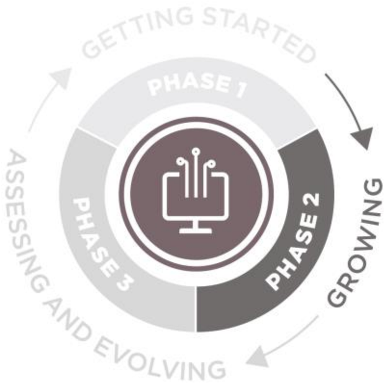
60-90 minutes

Activity Instructions

1. List the potential responsibilities (i.e. tasks and duties) for the position.
    a. If you already have a job description, you can take the responsibilities list from that.
    b. If you do not already have a job description, this is the place to go broad - not all these elements will make it into a final job description or value proposition.
        i. It can be helpful to set objectives for the number of ideas to be listed and the time to be spent, e.g. "Let's spend 5 minutes coming up with a list of 20 responsibilities / tasks / duties we think this job would cover."
2. Take a quick sticker vote to select the responsibilities that are the most critical / highest priority.
    a. 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 (e.g. most important, most likely, most interesting, etc.).
3. For each of your top vote-getters, discuss and capture how the responsibility / task / duty would solve problems faced by your application, program, and/or community or add new benefits. For example, could the person in this role:
    a. Save time and/or resources? Help the project increase resources?
    b. Improve community buy-in and engagement?
    c. Improve the application's quality or functionality?
    d. Eliminate risks the application, program, or community might face?
    e. Help end users use the application more effectively?
    f. Eliminate barriers to adoption?
    g. Create positive social consequences?

Activity 10: Value Propositions for Job Descriptions

4. Rank each of the elements articulated in Item 3 as “essential” or “nice to have.”
5. Finally, collate the “essential” value proposition elements into a value proposition document.
6. If you don’t already have a job description, you can use the top priority elements from Item 2 as the basis for creating one.
7. Share both documents with program leadership.



TECHNOLOGY

Phase II: Expanding and Integrating

Activity 11: Building Welcoming Communities

Goals

1. Identify what documentation your program has that supports growing a successful community
2. Prioritize missing documentation and develop a plan for its development

Prerequisites

Example: None

Who Should Participate?

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

Length

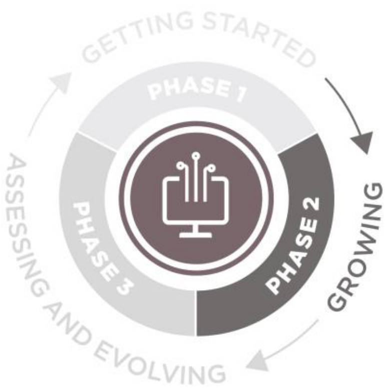
60-90 minutes

Pre-Work

Have a subgroup (2-3 program representatives) complete the checklist on page 2 and bring it to a larger group for broader discussion and prioritization. It is okay to determine that some components are not relevant or propose adjustments.

Activity Instructions

1. As a group, review the submitted checklist. Does everyone agree or not? Are there elements not included in this assessment?
2. Once general consensus on rankings is achieved, lead the group in identifying what components the group considers necessary to work on, then prioritize (as you cannot do everything at once). 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 working on the component - either taking on the responsibility for creating or improving it, or assigning the task to someone else and following up on its completion



TECHNOLOGY

Phase II: Expanding and Integrating

Activity 11: Building Welcoming Communities

Building Welcoming Communities Checklist

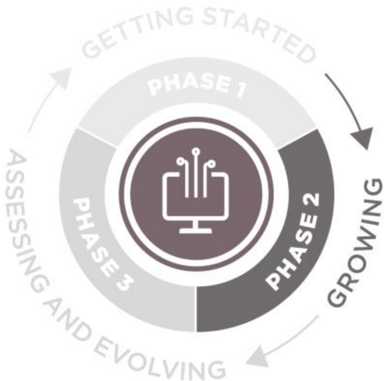
Does your program have the following documentation?

	Yes	No	In Progress	Unsure	N/A
A friendly README					
Clear code examples					
Code contribution guidelines					
Good first issue tags					
Response plan for new contributors					

Growing Your Community

How can you provide your community with a good foundation for community growth? Do you have the following?

	Yes	No	In Progress	Unsure	N/A
Code of Conduct					
Contributors or Authors file					
Regular newsletter or other communication thanking contributors					
Organizational repository (vs. personal)					
Structured roles for newcomers to fill?					
Clear and transparent communication channels?					
Regular schedule and structure for trainings, onboardings, workshop, etc.?					

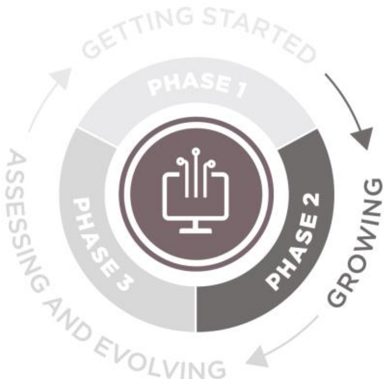


Activity 11: Building Welcoming Communities

Resolving Conflicts

Do you have a plan for conflict resolution as your community grows? Do you have the following helpful tools?

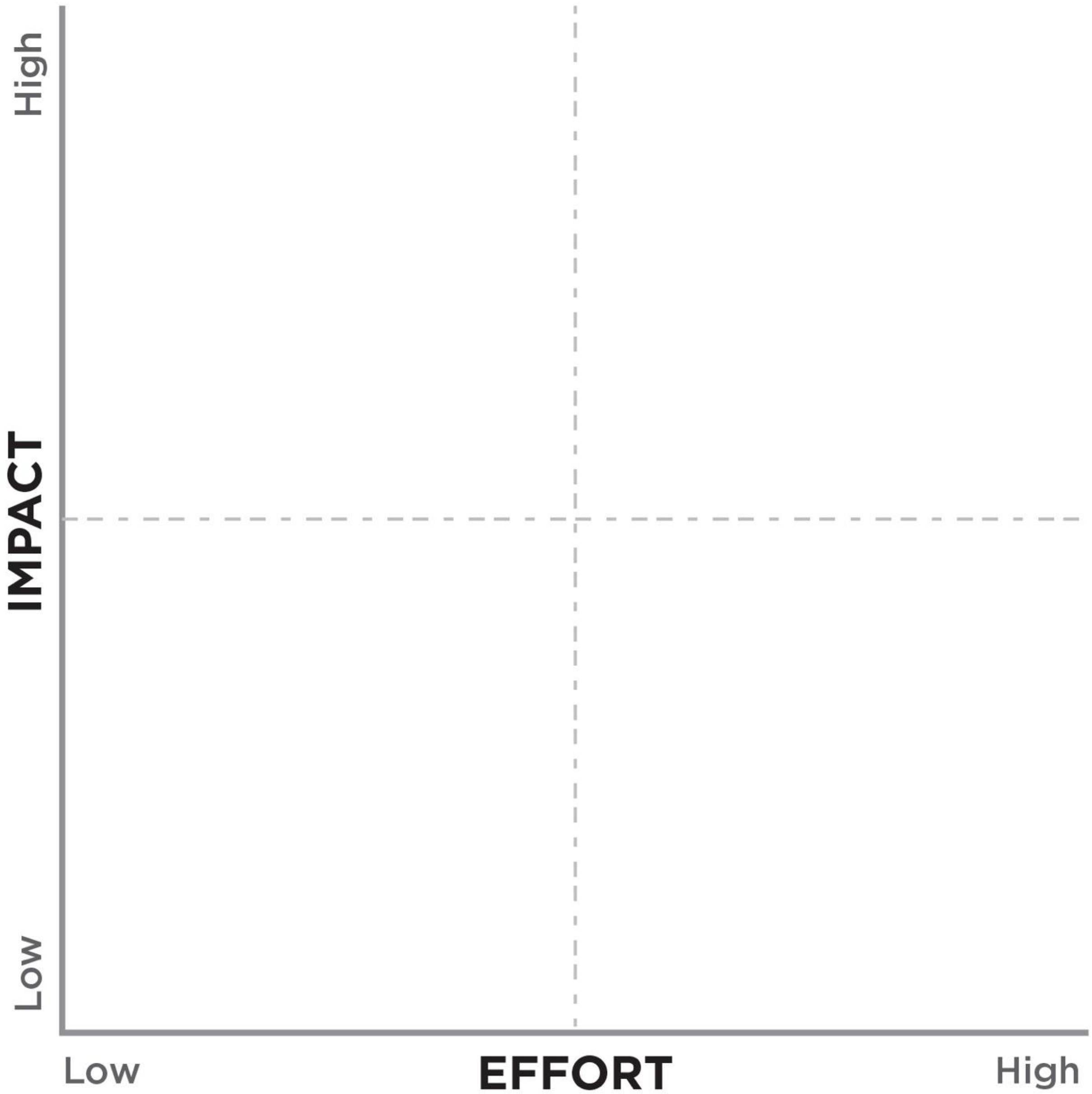
	Yes	No	In Progress	Unsure	N/A
Product vision and roadmap					
Documented decision-making process (e.g. consensus, voting)					
Identified community tiebreaker					

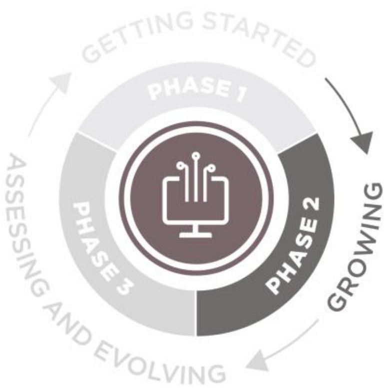


TECHNOLOGY

Phase II: Expanding and Integrating

Activity 11: Building Welcoming Communities





TECHNOLOGY

Phase II: Expanding and Integrating

Activity 12: Not Invented Here: Assessing Integration Options

Goals

1. Identify functional requirements or feature requests that may be solved by integration
2. Identify and assess potential integration solutions to satisfy functional requirements or feature requests

Prerequisites

Development Roadmap, Tech Activity: Landscape Analysis

Who Should Participate?

Program leadership (strategic thinkers), Program management (tactical thinkers). This activity is best kicked off with a small group together (in-person or virtually), and then moved to asynchronous work.

Length

Part 1, 60-90 minutes with a small group. Part 2, depends on the scope of the project.

Activity Instructions

Part 1: Small group together

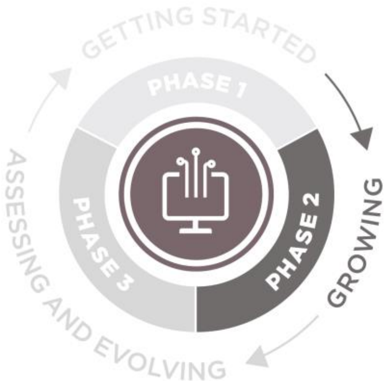
Identify functional requirements or feature requests on your roadmap that may be solved by integration. Sample questions to ask about proposed features are below; if the answer is yes to a majority of the questions, then integration may be a good idea.

Criteria	Yes	No
Is the request for a brand-new feature?		
Does the functional need have a timeline or expiration date?		
Is the functionality needed quickly, e.g. within one or two quarters?		
…		

Part 2: Small group asynchronously

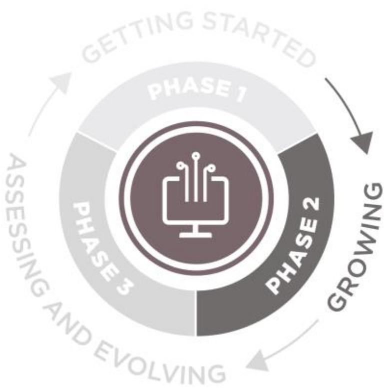
- Compare functional requirements with the features/functionality of the solutions identified in your landscape analysis
  - It may be useful to break the functional requirements into user stories to compare
- Assess suitability of potential solutions
  - Does the platform have a compatible license?
  - Is the identified solution designed as a component or amenable to re-use?
- Complete a gap analysis
  - Determine what is missing from your functional requirements
  - Prioritize the gaps with your community - are they must haves?

 Exercise adapted from “Overcoming ‘Not Invented Here’ Syndrome - Developer.Com.” Accessed December 22, 2020. https://www.developer.com/design/article.php/10925_3338791_2/Overcoming-quotNot-Invented-Herequot-Syndrome.htm.



Activity 12: Not Invented Here: Assessing Integration Options

- Map the integration
  - Create a workflow document that shows how the two systems would interact.
  - Is one system fully contained by the other, or do they just communicate? If the latter, does data flow both ways? Which is the system of record?
- Learn about and communicate with the other program
  - Who is their core community? Does it overlap with yours?
  - Who are the executive, strategic, and tactical leaders?
  - Are they open to an integration? Would their community benefit?
  - Do they have a sustainability plan?
- Determine deal breakers
  - What would stop an integration from being effective?
- Assess outcomes
  - No solution compatible - program will have to build
  - Multiple solutions compatible - assess them all to see which one is the most compatible
  - One best solution - work with governance and technology leadership to determine formal process for moving forward with integration (e.g. is an MOU or other agreement needed?)



TECHNOLOGY

Phase II: Expanding and Integrating

Activity 13: How We Retire Features

Goals

1. Identify elements of the platform that can be retired
2. Develop a plan for communicating retirement plans with the community
3. Communicate the retirement decision to the community

Prerequisites

Example: None

Who Should Participate?

Program management (tactical thinkers), Program staff (operational experience). This activity is best kicked off with a small group together (in-person or virtually), and then moved to asynchronous work.

Length

X minutes

Activity Instructions

Part 1: Small group together

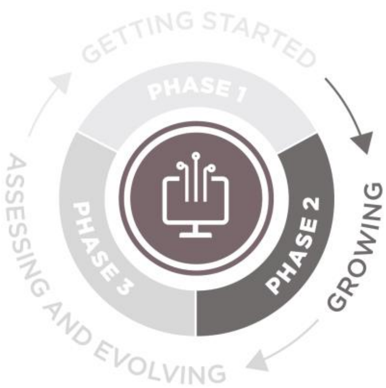
Note: If a specific feature is already under consideration for retirement, you may move directly to Part 2 of the activity.

1. Discuss/brainstorm together a list of features/functionality that may be suitable for sunsetting. Plotting features on an effort/impact matrix (sample on page 3) may help identify elements that are difficult to maintain but do not provide a huge value to users.
2. If there are several, select 1-2 features for further assessment.
3. Determine a timeline for the remainder of the activity, who will be responsible for its completion, and who will be accountable.

Part 2: Small group asynchronously

1. Gather and review data on each of the selected features:
   a. User data
      i. How many users will be affected?
      ii. What percentage of overall users use the feature?
      iii. Did they try it once or did they use it all the time?
      iv. What percentage of users *do not* use the feature?
      v. Why did those users not use it?

   b. Data on alternative workflows
      i. What specific workflows is the feature supporting? When asking users about how they're using specific features, beware the XY problem; or asking about a user's solution rather than the problem they're trying to solve.
      ii. What alternatives are there to complete the workflow?

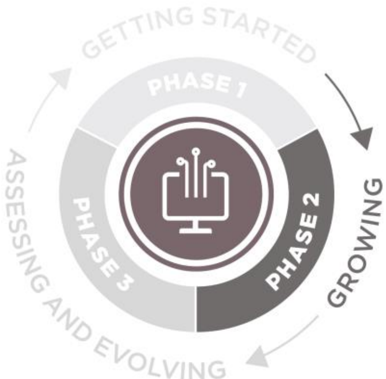


c. Data on organizational effort
    i. What resources (engineering, support) does the feature require?
    ii. What percentage of product use and/or revenue comes from the feature?
    iii. What is the expected cost of keeping it vs. the potential loss of removing it?
    iv. Is the product adding or distracting from our program mission?

2. Make a recommendation to retire/not retire features based on the data and information gathered.

If, after the above elements are complete, the decision is made to sunset a feature, move on to:

3. Gain approvals from the appropriate governance/leadership representatives.
4. Develop a sunset communication plan
    a. See Technology Activity: Developing an End-of-Life Communications Plan
5. Communicate
    a. Internal
        i. Ensure all program team members are aware of the change and the key details - why, when, alternatives
        ii. Update support documentation to clarify that a feature is no longer available, and what alternative workflows are available.
    b. External
        i. Reach out to affected users with a simple, concise message - enough detail so that nothing is unclear, dates and times the feature will be discontinued, and recommendations for alternative workflows.
        ii. Reach out via different channels: emails, in-application messaging if available, phone calls, social media.
        iii. Monitor feedback in the days/weeks following the announcement.



TECHNOLOGY

Phase II: Expanding and Integrating

Activity 13: How We Retire Features