Open Hardware Creators
in Academia Fellowship

Enabling Practices: Skills Inventory

Useful skills to successfully produce open source hardware in academy broken into five phases.

Phase 1: IMAGINE an open source development project idea in academia

At the earliest conceptual stage, consider how your OSH project would be received in academia: advocate for it from the beginning:

- Advocate for the concept of Open Hardware in your institution
- Link the open hardware goals to traditional academic goals (i.e write a paper, aligning with department goals, etc)

Include academic educational and research aspects at the earliest stage: including contributors, users and students. These include:

- Audience researvch / thinking creatively about additional use cases and users
- Connection to a community of users who are willing to help you develop an ecosystem, understand the community and strive for community buy-in

Think broadly: brainstorm on additional use cases, participants. Conceptualize while keeping open-source aspects in focus:

- Structuring your project to make it easy for others to contribute, understand the balance between hierarchical and flat organizations
- Understanding different role types which can be transferred from open source software (“Maintainer”, “core contributor”, “community manager”, etc) which might be critical for open hardware

OSHWA

Phase 2: INITIATE

Evaluate existing resources (PEOPLE, SPACE and TOOLS, FUNDING), and the need for new resources. Apply for funding if needed.

People

- Engaging students to distribute workload/research plus proper mentoring, follow-up
- Interdisciplinary collaboration builders

Space and tools

- Fabrication or lab space / test environment (this includes testers/ need feedback, prototyping), toolset
- What can be done in-house, what needs to be outsourced?

Securing funds - approaches differ for high-risk versus standard OSH projects

- Getting sponsorship from relevant stakeholders in your university/area
- Money (grant writing, beyond skills), team (depends on scale of project)
- Obtain funding for creating open hardware by coupling to grants that need new functionality (e.g. not available commercially or too expensive)

Phase 3: IMPLEMENT

General skills needed:

- Project management
- Understand practical limitations both in time and in resources
- Documenting meta aspects of the project for reviewable design, development, and sustainable maintenance
- Communication skills, e.g. not speaking in acronyms to be inclusive
- A clear plan for dissemination for response/feedback
- Documenting for different audiences (academics, students, final users)

OSH-related skills:

- Familiarity with open source building blocks, e.g. Arduino, Design Software (kiCad) – avoid behind the paywall, but verify sustainability of elements you rely on
- Familiarity with open source documentation interfaces (eg git)
- Workflow - having foresight to know which components of doing open hardware work need to be documented at a given stage in order to avoid dependency related problems
- Publish the designs in open hardware journals like HardwareX and JOH and then publish using the free and open source hardware (FOSH) in more conventional journals – for every project

Phase 4: SUSTAIN

Imagine problems that may arise for new/outside users, be open to suggestions/ideas from others:

- Good communicator to diverse audiences
- Having a zeal for documentation, maintaining documentation
- Provide troubleshooting documentation or feedback to users
- Understanding where UX is best as diagrams, explanatory notes, and heavy text for ease of recreation or derivatives.
- Willingness to work in the open (sharing unpolished steps, etc)
- Having a good sense of what matters most to the given audience, and always give polite feedback

Phase 5: SCALE

Support inclusion of additional technical elements, additional use-cases, and cost saving possibilities (if exist):

- Avoid using materials or parts you cannot obtain readily in quantity if you plan to scale
- Technology translator, ensuring many communities can understand your project
- Giving the possibility for others to better the original design (git reviews, merges)

Route to commercialization; evaluate open source aspects of intellectual property issues:

- Evaluate intellectual property licenses and where they can be used, both in source and geographical boundaries
- Knowing who to consult in case of licensing or applying standards - who’s the tech transfer office equivalent for open source hardware?
- Have partners for commercialization / Finding partners to scale out/up
- Decide what, if any, plan you will do for order fulfillment /shipping if you are planning to sell or distribute your hardware directly