How Sakai Development Works

Projects and the Released Product

Many activities in the Sakai community do not themselves produce code. They can range from sharing pedagogical practices and user research to QA testing and documentation. This is all valuable community support, even where it does not have a concrete outcome in the released product. At the same time, many development projects that do produce code are not necessarily included in Sakai releases. We think this is a sign of health. Sakai needs R&D efforts, for one, while there may also be many niche applications or self-contained projects that could benefit from their own release cycle. The Sakai product should be a platform that supports all this. The end result is that the effort that goes into an official release represents only a fraction of community activity.

Official Sakai releases call for a special level of coordination and quality assurance. Above and beyond satisfying user requirements, Sakai releases must offer:

- a coherent, accessible user experience
- proven technical stability and scalability
- code that is maintainable through consistent practice and shared ownership

The Sakai community uses a process designed to ensure that what does go into a release meets these standards. Doing this effectively involves a significant amount of planning and consultation through the entire development cycle.

The Product Council

The formal stewards of the Sakai Development Process are the members of the Product Council. The Council fulfills its formal role by establishing through objective criteria which projects are ready for a release, and informally by advising projects as they progress from R&D to production-ready maturity. The Product Council will undertake its work:

- by employing the expertise of its members
- through direct consultation with experts in the community
- with reference to best practices for technology, pedagogy and standards
- by establishing and communicating clear and objective criteria with a transparent process that allows review and comment from the community

Why a Project would want to go through the Process

The motivation to start a project usually comes from a strong local need or interest. There is a natural reluctance to expose one's project to external review, risking extra effort in discussion or re-design as one attempts to turn a local inspiration into a general solution for the community. Given that understandable concern, it’s worth saying a few words about what we think development teams should get out of the product development process:

- Sustainability and maintenance support: at the end of the product development phase the code should be ready to be handed off to a maintenance team, so that developers committed to the initial project need not be devoted to maintaining the code indefinitely. By the same token, institutions should be able to reclaim their development resource for further projects.
- Help attracting complementary resource: many local efforts do not have access to a full cross-functional team - ranging from designers and functional experts to system architects - to produce robust solutions on their own. The Product Development Process should help surface contact points for additional contributors, and provide some support in recruiting them from elsewhere in the community.
- The quality that comes from expert review: the expertise that the Product Council both contains and can call upon should help guide projects to an end result of higher quality, and presumably even greater satisfaction of the local need.

The aim is not to impose onerous barriers, but rather to help Sakai projects succeed.

Characteristics of Strong Projects

- Led by user-centered design. This should include a design effort that precedes coding, designing for accessibility from the beginning, iterations of user testing, and in general a plan attentive to user requirements.
- An open development process that engages deeply with the Sakai community early.
- Shared ownership. Since local priorities can change, or individual developers may move on or be reassigned, it can be dangerous to rely on a single developer or institutional commitment. There may be exceptions, but a good rule of thumb is that risks to sustainability are tolerable if at least 3 institutions are committing resource to the project.
- A project plan with scheduled milestones, opportunities for community input and contingency plans for major risks.
- Consistency with Sakai coding conventions, ranging from appropriate use of common frameworks and services to code style and formatting. This has a bearing on both the quality of the code and its maintainability.
**Stages of the Process**

**R&D**

This is not a formal stage of the process, but it's included here for completeness. Every effort starts in R&D: there is no barrier to entry, nor is there any assumption that an R&D project intends to become part of a Sakai release. This stage is where new ideas are generated and new technologies are tried. Projects may emerge from R&D and go on to mature independently of formal Sakai releases.

To go on to Incubation, the project lead(s) simply need to supply basic documentation about the project, its participants and aims.

**Incubation**

**Criteria:** Incubation Documentation

Incubation is the first formal stage for projects that intend to be included in a Sakai release. The goal in the incubation stage is to prove desirability to the community, formalize project requirements, assemble a cross-institutional development team that includes functional expertise, build a project & maintenance plan and reduce development risks. We expect formal structures to ensure the reliable allocation of resources and that the operational decisions during the project are driven by end-user priorities. The Product Council plays an advisory role in the shaping of this plan.

To go on to Product Development the project must show a strong plan and adequate resources to deliver.

**Product Development**

The product development phase is essentially executing on the plan developed in incubation.

To go on to Maintenance (i.e. inclusion in Sakai releases), the Product Council must agree that it is ready, and the maintenance team responsible for releases must also agree that it is maintainable. It will then be included in the planning for the next Sakai release.

**Maintenance**

This phase is for bug fixes and minor feature enhancements. This activity should generally not require the involvement of the original project team, and the released code may be iteratively refined without needing to go back through the product development process.

**EOL**

Some capabilities may be superseded over time, or may simply no longer be maintainable, and will need to be retired in a way that doesn't disrupt the Sakai community unduly.

**Deprecation Policy**

---

**Sakai 2 vs. Sakai 3**

At this juncture of Sakai's history, at the advent of the Sakai 3 generation of product, it's worth saying a few words about how the criteria may differ between the generations. Some standards are far more expensive to retrofit than to plan for from the beginning (e.g. accessibility), while Sakai 3 will also be less forgiving of idiosyncratic design and functionality, insofar as it inclines toward cross-functional activities and workflows. The net result is that Sakai 3 projects will in some respects be held to a different standard, one which would be impractical to apply to certain portions of Sakai 2 code at this stage.