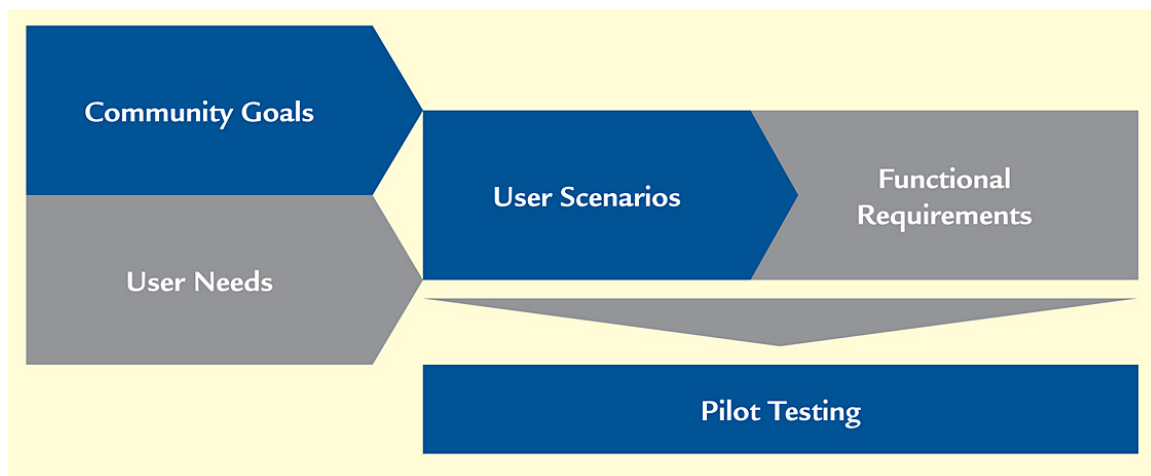


Technology for Online Communities of Practice

As *Connect and Inspire: Online Communities of Practice in Education* makes clear, launching a successful online community of practice is much more than a technology project and requires a thoughtful strategy that considers the community’s goals, incentives, roles, content, and many other nontechnological criteria. That said, the technology must support the types of valuable interactions that will make participation worthwhile for community members. Because there are dozens of collaboration platforms and tools, each with different permutations of features, choosing the configuration can be overwhelming. Countless collaboration initiatives have failed after extensive requirement-gathering and platform-evaluation efforts that focused on technology—at the expense of goal-setting and planning for the activities the technology needed to support. It is far too easy for the technology component of a collaboration project to take on a life of its own and miss the big picture. The solution is to engage in some careful planning but also start small, focusing on a narrow audience and set of features. We explain this approach in more detail in what follows.

Community Planning

To avoid the pitfalls endemic to online community projects, the technology evaluation process must never lose sight of overall goals of the initiative and how the platform will create value and connections for proposed participants. Doing so requires an iterative process to derive user scenarios and functional requirements, against which supporting technologies can be vetted. The diagram below shows a workflow for a community platform evaluation:



Community Goals

A technology evaluation effort should begin only after community leaders, sponsors, and stakeholders have clearly identified goals for the community. This means there should be consensus around how the community will create value for its membership and what the beneficial outcomes must be after one year, five years, and beyond. Any effort to begin such an initiative without this focus sets itself up for failure. Note also that the community's purpose must be clear enough that potential members can grasp it and understand why they should become active members. The set of goals thus becomes an important member touchstone that guides participation and growth.

Audience Needs

Along with goals, community initiators also must understand the needs of their desired members. The community's purpose will determine the kinds of individuals the community will want to recruit and attract. The prospective community manager must develop a deep understanding of what these audiences require and how they behave online. When it comes to determining what functionality a technology platform should provide, successful online communities of practice tend to focus on doing the few things that actively engage all segments of their memberships and on avoiding additional tools and features that do not meet this crucial test.

Audience Scenarios and Functional Requirements

The key question, then, is how to translate the high-level objectives of a community and the needs of its audiences into a set of clear functional priorities that inform technology selection and configuration. Developing a set of user scenarios, which are concise statements describing tasks or actions that users will wish to perform in the community, is a successful approach. The scenarios might describe several steps for a user to get from point A to point B to achieve a certain informational or transactional goal (e.g., find and download a useful resource; discover, read, and comment on a post; or search for and connect with a peer working on the same topic). These need not be especially formal or exhaustive, but they should be done well enough that those planning the community capture all the key illustrative stories pertinent to the audiences in question.

Wenger, White, and Smith (2009) develop a list of common orientations that communities can have, one or more of which should emerge in the scenarios. These are as follows:

- **Meeting support**—Some communities emerge from one or more meetings. A face-to-face meeting may spark follow-up online gatherings or a series of meetings may deliberately entail a blended approach, having a mix of online and offline sessions. There may even be fully online meetings. Online, meeting-oriented communities may engage in presentations, sharing of materials and information, or decision making.

- **Open-ended conversations**—Some communities are represented best by their archetype, the classic discussion group. There may be face-to-face meetings, but the community is primarily built upon a series of ongoing online conversations.
- **Project infrastructure**—Some communities form around a project, with an online community serving as one piece of the overall project infrastructure. These communities are focused on sharing key information, creating artifacts, and solving problems of practice.
- **Content collection and organization**—Communities have formed around the task of collecting and categorizing information to make it useful for members to find and access. In some cases, the community also develops syntheses of the information. These communities obviously rely on repositories for files and links of various forms and the ability to categorize them in ways that make sense to community members.
- **Access to expertise**—Although knowledge can be useful in the form of collected content, there is often a need for finding the right person with the right expertise. Communities can form around this orientation, presenting experts and making them available to its members for questions and requests.
- **Building and maintaining relationships**—Relationships between people matter, and some communities orient themselves to building and maintaining interpersonal relationships. They may do this for explicit learning, or they may just want to maintain a network for personal or professional reasons. These communities may be open, but they often are closed, with members caring deeply about who becomes a member.
- **Individual participation within a larger whole**—Communities can become spaces where individuals can participate quite differently from one to another. There is a higher level of personalization and asynchronous activity support in these communities than in others.
- **Community cultivation**—Some communities have a set of members who actively care for content. They mine community content for valuable information and repackage it in easily accessible forms. These communities require active governance and accepted leadership to facilitate the work.
- **Context-serving**—Some communities are distinct in that they serve a clear mission that gives them unique identity. They may do this because they bring people from different organizations together to focus on a particular problem. They may also exist within a single organization. Some may be open and some may be closed. They may take the form of any of the other communities described here, but the context in which they work stands out as a key feature.

Another way to think about scenarios is to consider the levels of interaction that are desirable, the kind of knowledge development community leaders and stakeholders want to occur, the kind of resources they can put toward the venture, and the characteristics of the desired members of the community. Possible levels of interaction are as follows:

- **One-way dissemination**—Some communities primarily provide a one-way dissemination of useful knowledge or resources from a central knowledge producer (or group of producers) to participants who are primarily knowledge consumers. Participants might post comments about the knowledge being disseminated, ask questions, or post suggestions for what else would be useful, but the flow of knowledge is primarily in one direction.
- **Shared development**—Communities that support a group of users working together toward some end—such as planning an event, writing a document, managing a project, and so on—build knowledge in multiple people at once (though not always at a uniform rate) as each creates something new, more or less in parallel.
- **Many-to-many interactions**—Some communities support many-to-many interactions among participants. The knowledge might still flow primarily from a small pool of active users to broader pool of participants, but the users tend to be peers, whether knowledge producers or consumers. This type of online community interaction is the most difficult to foster, because it often relies on volunteerism among the participants to take the time to contribute ideas, questions, opinions, and so on. Motivating users to actually post comments or contribute knowledge is much more difficult than motivating users to consume useful knowledge posted by others.

Both these lenses can help identify relationships between and fill gaps in the collection of audience scenarios that will guide technical implementation.

Once those developing the community have identified audience scenarios, they can prioritize them against the overall community goals and the target audiences' needs. The process proceeds in this way:

1. Map discrete audience needs to corresponding audience scenarios (e.g., if an audience is school technology administrators who need to understand mobile devices, matching scenarios might include ones for accessing a knowledge repository and submitting questions to experts).
2. If not already explicitly done, assign an importance (1–3 scale) to each audience and each stated audience need.
3. Use these importance ratings to prioritize the list of scenarios, with those scenarios mapped to the most important audiences and needs rising to the top.

4. In parallel, assign similar importance values to the community goals. Those scenarios that most clearly support the most important goals would gain additional priority. Although these quantitative weightings provide a good first step, in almost all cases, the prospective community manager must layer some subjective analysis on top of the scores to arrive at the optimal set of functional priorities.

It is always tempting to create a large list of priority scenarios—and therefore features—in order to address the full range of problems that may exist for the target audiences. Those developing a community should take a hard look at the final set of priority scenarios and apply a critical eye to them. In addition to being true to the goals and user needs, they should represent a small number of discrete, well-connected activities (and thus lead to a small amount of community functionality). These activities should provide sufficient variety to engage potential community members with different preferences while simultaneously not spreading member attention too thinly or overwhelming new members with options. As *Connect and Empower: Online Communities of Practice in Education* notes, communities will be more effective to the extent that they try to do a small number of things well rather than provide solutions—often ultimately mediocre—to a large number of potential problems at their launch.

The same is true, by the way, for existing communities. A community that is failing to see activity and growth might be trying to address too many audience scenarios at one time by offering too many different features. It should review the scenarios it is trying to address, reprioritize in light of goals and audience needs, and turn off technical elements related to lower priority items.

Functionality and Pilot Testing

With a reasonable set of prioritized scenarios in place, the prospective leaders of the community can define the technical functionality that matches the scenarios. Again at this level, those building and stewarding a community often are tempted by the notion that more functionality must be better. In fact, superfluous functionality actually detracts from the core-value-creating activities. This is especially true for communities targeting professional audiences, who tend to be busy, impatient with technology, and already suffering from information overload related to their professional responsibilities. They want easy access to the one or two things that will help them get information or knowledge they need to do their job. (e.g., “What are the materials I need to participate in the next meeting?” or “Who are the three people I need to contact to get answers related to the issue I currently face?”)

The choice of functionality has a great deal to do with the community’s purpose. In a community, for example, whose key value proposition is allowing a small group of “knowledge owners” to share their experience and wisdom with a large group of “knowledge consumers,” the appropriate tool might be a blog. This format allows the knowledge owners to post useful knowledge that is easily consumed and commented

upon by the community. A peer-to-peer discussion forum for the knowledge consumer participants might fail if all they want to do is read the posts by the experts. A second community, whose primary value is connecting members working on the same topics and encountering the same problems and questions, might be best served by facilitating discussions, allowing users to post and answer questions, quickly share resources or observations, and respond accordingly. Social networking functionality, such as rich profiles, friending, and messaging, might also be useful in this scenario. On the other hand, also offering member blogs for such a community might actually detract from the critical mass of the discussion area as users' questions, ideas, and conversations are now split between two separate places. Last, in neither community described above would a wiki work well because this tool tends to be better suited to organized groups creating a single and coherent body of knowledge, but wikis often fall flat in less structured communities that facilitate ad-hoc, ongoing knowledge sharing.

It makes sense to start not only with a small feature set but also with a rollout limited in the scale of its community and its technology. In short, community leaders can be well served by beginning with a limited pilot phase. The history of online community initiatives clearly shows that, once activated, community participants will take a community to where it makes the most sense for it to go—regardless, sometimes, of initial intention. This is a natural process and should be honored; if the end goals remain in line, community initiators should allow themselves to be pleasantly surprised by different means to the same end that can emerge. These different means may imply adding (or subtracting) technology features and making other refinements that enhance the community's chosen direction. In addition, tracking the emerging patterns of activity may suggest new functionality that could improve the usability of the community. If it turns out, for example, that community members are regularly accessing image files from discussions, adding a photo gallery (or incorporating a feed from a service like Flickr) may be a valuable addition. Launching a pilot-level community first can allow these “in practice” directions of the community to emerge without a large investment in a different technology direction (in addition to the standard working out of kinks).

Community Management

A final overarching point is in order. Those launching a community should be under no illusions of the time the effort will require of them. It is true that many successful communities have emerged organically, but these required significant effort on the part of those who grew them, both in providing the content for the community and in developing, at least informally, the norms of community interaction. More than an offline community, for instance, users expect a moderate amount of existing content from a community before joining. This content must be seeded. Within the technology structure, the direction and norms of the community must be explicitly and implicitly clear. This requires deft user experience configuration and wording choice. Training many users on new technology is an important, but large, burden. Community managers should not only provide users with static help (e.g., FAQs and “cheat sheets”) but also

do regular webinars so that new members can more clearly understand how to use the tools and effectively contribute to the effort (though too many questions about the same functionality generally indicates there is a design issue with that feature). Growing a community where none exists requires the additional effort of recruiting participants and bringing them into the fold. There will be ample work required in community moderation and marketing. Managers must keep online discussions active, welcome new members with a post or e-mail, and cull news for content ideas. Finally, managers must watch system metrics to understand how members are using tools and what content appears to be most valuable. This analysis will help make decisions about further technology investment. In general, expect that the community will not grow without a significant time investment.

Community Tool Functionality

This section outlines different types of community functionality that may be valuable.

Content Tools

Content management and file repositories—Many communities of practice revolve around sharing and disseminating knowledge artifacts. Community platforms need to mix more traditional Web content management with social networking or other forms of collaboration. Variations run from simple file lists with folders to more complex hierarchical listings with metadata-driven filters. The amount of content a community might hold is obviously a primary driver in the choice of needed features. Larger amounts require a greater ability to organize and browse content. Size of community and familiarity of members with each other also makes a difference here. Smaller online communities reflecting existing offline networks often can manage a more utilitarian offering than one with a larger, less established network. *Examples: Any of the major “all-in-one” community platforms such as SharePoint; Central Desktop; Jive; and Drupal Commons.*

- **Blogs**—Blogs are a great tool for facilitating discussions when users wish to do more than just ask and answer questions. Blogs allow one user to post a brief or extended idea or opinion and invite others’ comments. The structure thus lends itself to a case where the community needs one or a few individuals (the authors) to have a priority position relative to the rest of the community members. Facilitating active group blogs may require an intuitive interface for posting that allows easy inclusion of images, video, and other multimedia. In addition, blogs should offer a user interface that highlights the authors, making it easy to see their profile and picture. This gives users a clear sense of the relationships they can make and the expertise they can tap. *Examples: WordPress; TypePad; BlogSpot; and many others (including “all-in-one” platforms such as Drupal Commons and Central Desktop that include blogs as part of a larger feature set).*

- **Microblogs/Status Updates**—Microblogs, which are functionally equivalent to status updates, emerged from the desire to take the “anyone can publish” ethos of blogs and apply it to short thoughts, quick observations, a shared links with minimal explanation. Twitter is the best known example, with its famous 140-character constraint on expression. This has spawned a litany of jokes, poems, and even mininovels, but it has also become a place for serious community interaction. *Examples: Twitter; Tumblr; also increasingly a common feature of larger “all-in-one” platforms such as Jive and Drupal Commons.*
- **Wikis**—Wikis are special websites that make it easy for users (or selected users) to add and edit pages as part of collaborative content development. Although wikis are good in some situations such as developing conference agendas, developing software documentation, and compiling frequently asked questions, the lack of preexisting structure often makes them chaotic or confusing for novice users. Although there are exceptions, posting content to a wiki often works better for more sophisticated users of the Web and for those with deeper content expertise. Some participants are hesitant to click on a button that says “edit,” and, if they do, often change their mind and decline to post something when an entire page opens up and is editable. Other tools that allow a user to simply post a single question or upload a shared resource tend to require fewer clicks and are less intimidating to novice users. *Examples: MediaWiki; SocialText; PBWorks; Confluence.*
- **Collaborative document authoring tools**—The best known example of such a tool is Google Documents. These tools work like a word processor, spreadsheet, presentation tool, or other familiar software in many ways, but multiple users can edit documents simultaneously with all edits tracked for reference. This feature can greatly ease collaboration as well as provide immediate feedback to a principal author(s). Less advanced tools provide a check-in/check-out capability so that versioning is clear. *Examples: Google Documents for simultaneous editing; “all-in-one” community tools like Basecamp and ProjectSpaces for check-in/check-out.*
- **Social bookmarking**—The ability to collect links from across the Web is an important feature for online communities of practice. There is great deal of content that has already been produced and just needs to be brought to a single place where members can access it. Social bookmarking services allow leaders as well as members to collect links to external Web content and categorize it. Newer services also allow annotation and commenting that displays on the page when other logged in users of the bookmarking service go to a linked page. *Examples: Diigo.com; Delicious.com; StumbleUpon.*
- **Media libraries and albums**—In many cases it is valuable for a community to share multimedia—photos and videos, primarily—in order to both capture knowledge and build camaraderie. Some community tools provide a means to do this within the application in a way that is well suited for the medium in question. Even when it is possible to include this content within a tool, it still can

be valuable to use outside services as the “home” for the content and embed the players they provide within the community platform. The embedded player provides an excellent user experience and the community leader does not have to worry about hosting the much larger files. *Examples: Flickr; YouTube; some “all-in-one” community tools like Ning and BuddyPress; Facebook.*

- **Data visualization tools**—When communities use data and want to share information and knowledge within that data in simple ways, they should consider collaborative data visualization tools. Data visualization tools can make data “come alive” and thus make it easier for members to access, use, and share that data across knowledge levels in a community. *Examples: Google Spreadsheets; Many Eyes; Swivel.*

Member Interaction Tools

- **Profiles and social networking**—For large communities, strong social networking functionality (that is, indicating who in a community knows whom and allowing users to follow the participation of others they know) is a valuable way to build, deepen, and maintain relationships (Cambridge, 2010). Members connect—or reconnect—with contacts in their professional field or personal contexts and create new avenues for interaction with these people. This is especially true for social networking systems such as Facebook, where the profile includes not just biographic information but an entire history of contributions (including The Wall). A rich profile feature also can be important for reducing barriers to participation and helping members be more comfortable interacting and sharing with people whom they may have never met in person. This can be a more important element of topical communities where users are seeking knowledge. *Examples: most “all-in-one” community platform tools like BuddyPress and Ning; LinkedIn for professionals.*
- **Member commenting**—Many online communities deploy user comment threads on other content as a primary means of interaction. Tying comments to wiki pages, documents, and blog posts allows rich discussions around all content. Providing this facility across all content leads to an increase in both the quantity and quality of user interactions. Advanced commenting tools allow rich threading, easier access, and the ability to track an individual’s comments across multiple communities. *Examples: All blog platforms; nearly all “all-in-one” community platforms; Disqus (to cross multiple communities); Omeka.*
- **Discussions**—Most communities allow members to engage in meaningful discussions to ask questions, test ideas, invite input, and share lessons learned. Unlike blogs, where there is an implied hierarchy of author(s) over others, discussions are generally set up with all participants as equals (though it can be valuable to call out some participants as being “special,” e.g., if they represent the sponsoring organization or are known experts). *Examples: Many “all-in-one” community platforms; social media tools like LinkedIn and Facebook; phpbb; ezboard.*

- **Webinar services**—It can be valuable to intermingle asynchronous online activities (in which members can participate at any time) with synchronous, virtual interactions (which require members to be online at the same time). Sometimes a conference call is sufficient for this but, in other cases, adding the possibility for users to share a screen and view the same Web tool, presentation, and so on in real time provides a more powerful and efficient means to get work done. These tools are commonly stand-alone services that go alongside other community tools. *Examples: WebEx; Adobe Connect; Elluminate Live; Dimdim.*

Member Feedback and Research

- **User-generated ratings and popular content**—Many communities allow members (and, perhaps, other users) to rate the utility of content, comment on it, and find things according to the ratings and usage of their peers. For sites that present a lot of content, ratings can be a powerful tool for users to quickly find the most useful content and understand how it is being applied by other members in the network. It also helps to highlight what is energizing a community at that moment. *Examples: Ratings and reviews on Amazon; eBay and other e-commerce platforms; Facebook’s “Like” feature. Some “all-in-one” community platforms now incorporate this feature.*
- **Polls and surveys**—These tools are excellent ways to facilitate participation from a broader group of participants. Some users are more willing to respond to a poll or survey than they are to post to a discussion forum or add a comment to a blog. They are also an excellent means of facilitating participation and involvement from larger communities, which often have a greater “fear factor” for posting ideas that many other people can see. This fear can create an ecosystem of a few active participants who dominate the discussions and a larger group that is hesitant to contribute. *Examples: SurveyMonkey; SurveyGizmo; Wufoo; polls included in some “all-in-one” community platforms such as Jive and Drupal Commons.*

Project Coordination

- **Event calendars**—It can be useful to have a central listing of events so all members can understand upcoming activities and find information from past gatherings. *Examples: Many “all-in-one” community platforms, including Central Desktop and Ning; MeetUp for online access to communal offline events; Evenbright.*
- **Task-management tools**—A more project-oriented community will want project-management tools such as task lists and milestone tracking. *Examples: Basecamp; Central Desktop; ProjectSpaces.*

- **Decision-support tools**—Groups collaborating on projects must develop strategies and make decisions. Decision-making processes can be supported by tools for ranking ideas, establishing consensus, or systematically analyzing a situation through a structured set of steps. *Examples: Decision-support tools are offered within some project-management suites, such as SAS Streamworks, and through online idea-ranking and deliberation systems, such as IdeaScale and HERMES.*

Incentive and Recognition Services

- **Badges**—Indicating levels of stature or achievement within a community can help motivate participation and help members identify experts. This is often done by allowing users to earn, give, and receive badges indicating some accomplishment. The badges become part of the user’s profile. *Examples: Edmodo.com; customized versions of BuddyPress and DrupalCommons.*
- **Rewards systems**—Though still emerging, a step beyond badges is the ability to earn physical rewards for actions within a community. Obviously, this increases costs but could increase incentives in some situations. *Examples: uBoost; Webcentiv; and Love2Reward.*
- **Reputation management systems**—As people become more and more connected online, it is useful to know whether the person making a particular comment is reliable. Reputation systems that assess a user’s behavior and gives other users the sense of whether the user is an active participant, whether others have responded favorably to the user’s contributions, whether the user has a history of acceptable discourse, and so on. In that way, a reader of a piece of contributed content in isolation gets the added context of the author’s general reputation. *Examples: eBay and Amazon “rate this seller” tools; Slashdot and other community news sites; customer support “was this valuable/helpful” features that roll up to the author.*

Social Media and Community Sites

- **General social networking sites**—Online communities these days have blurred boundaries. Because of the proliferation of social networking sites, particularly Facebook and LinkedIn, community designers need to decide which activities exist within sites branded, in some way, as the organization’s own and which are pushed into third-party sites. Some organizations choose to only focus on third-party sites, although the community functionality of these sites is limited.

Many communities now coexist with the social networking sites, with core activities occurring “on domain,” but with some conversations and activities extended into Facebook and LinkedIn. A number of platforms support automated (or facilitated) linkages to the social network sites. In other cases, versatile APIs offered by the major social networking sites combined with simpler options for building widgets allow custom integration. And in some

cases, community managers choose to inhabit both “on domain” and “off domain” communities in parallel efforts, acknowledging that the third-party sites represent “where the people are” and, therefore, need to be included into the discussion of community strategy, design, and platform.

- **Topic-specific community sites**—In some cases, developers have built platforms that are specific to a topical sector (such as education) and which offer many of the features of more general purpose platforms. These can function as any other community platform if the specialized design serves the new community’s needs and if they are open to the audiences that the new community seeks to target. They may also include features that are needed within the context of the domain, such as greater privacy and control for social networking in a school environment. In some cases, using such a community may provide a strong way to recruit members, because individuals may already participate in these systems in other contexts and be comfortable with them. These services are also likely more tailored to specific audiences and may offer higher levels of security and privacy. *Examples in the education sector: edmodo.com; Edublogs.org.*
- **Content-type-specific social media sites**—A number of sites have emerged that allow social sharing of specific kinds of content. YouTube is the best known example, where the content type is video. These can be valuable services for disseminating media content and encouraging communal interaction around it. *Examples: YouTube and Vimeo for video; Flickr and Picasa for photos; SlideShare for presentations; Scribd for documents; Swivel and Many Eyes for data.*

Supporting Utilities

- **Notifications**—Notifications are critical components of an active community. Although some complain they get too many notifications, e-mail is still valuable to many members and remains the primary way most users are notified of activity in communities where they participate. E-mail notifications allow busy people to stay informed about new resources, discussions, and other activities without having to log in each day to check. Some users prefer e-mail and other, more sophisticated users might prefer content syndication via standards such as RSS or Atom. SMS notifications also may prove important for some users, as may notification via social media tools such as Twitter. *Examples: Many “all-in-one” community platforms, including Basecamp and Central Desktop.*
- **Public facing and password-protected content**—For some groups, much of the community content, such as sharing best practices, lessons learned, and tools and resources, could be on publicly accessible pages. Putting content behind a

password raises the bar for participation and may deter some users from engaging in the community, but it also provides a level of trust and feeling of privacy. Different tools offer different levels of ability to make some content private and some not. In some cases there is a broad brush: this whole “site” is private, that whole one is not. Others offer the ability to make some individual sections, or even content items, public and private. *Examples: Essentially all “all-in-one” community platforms, with Jive and Drupal Commons allowing more granularity; Facebook.*

- **API**—An Application Programming Interface (API) provides a set of rules and specifications to allow one Web-based program to interact with another program. The proliferation of APIs—one of the most consequential recent trends online—allows for “mashup” applications, which pull data and functionality from multiple sources online to combine them in new presentations. Most users see the results in Facebook and mobile phone applications that enable presentation and updating of Web content outside its “native” context. *Examples: Ning; Facebook; Twitter.*
- **Centralized authentication**—Websites increasingly allow user authentication and login via Web-wide authentication services. The advantage of offering such a feature is that it eases a barrier to joining the community—the prospective member does not need to create yet another account they have to remember. Access to this feature is limited by the platform provider, in general, though the option is becoming more common. Available open standards are OpenID or OAuth, as well as extended authentication from the major sites such as Facebook, Twitter, and Google. *Examples: Google/OpenSocial; Yahoo!; and OpenID, as supported by extensions to many platforms.*
- **Metrics tools**—Web applications increasingly offer their own set of Web metrics tailored to their specific activity. They are useful for understanding the way that tools are being used and by whom they are being used. These can be supplemented by generalized Web metrics services. *Examples: Built-in tools for each service; Google Analytics; Omniture.*

Community Architecture

Community sponsors must merge desired and available functionality together to create a coherent service. Note that “coherent” does not necessarily imply “single solution.” Often a single solution is not even possible. Consider how the community orientations offered by Wenger, White, and Smith (2009) map to possible functionality (and note that there can be variability within some cells based on the specific nature of a community).

Key ▲ Important ● Nice to have	Community Orientations								
	Content-Oriented			Relationship-Oriented			Task-Oriented		
	Content	Community Cultivation	Access to Expertise	Open-Ended Conversations	Relationships	Individual Participation	Meeting Support	Problem solving*	Project Infrastructure
Content management and file repositories (incl. media)	▲	▲	●	●	●	▲	▲	▲	▲
Discussions	●	●	●	▲	●	▲	●	●	●
Blogs	●	▲	▲	●	●	●	●	●	●
Microblogging	●	▲	▲	●	●	●	●	●	–
Wikis and other collaborative authoring tools	▲	●	–	–	–	–	●	●	▲
Webinar services	●	▲	●	●	▲	●	▲	▲	▲
Social bookmarking	●	●	●	–	●	●	–	●	–
User ratings and popular content	▲	▲	●	●	●	▲	●	▲	●
Polls and surveys	●	▲	●	–	●	●	▲	●	●
Event calendar	●	●	●	●	●	●	▲	●	▲
Task management tools	●	●	–	–	–	–	●	●	▲
Badges, reward and reputation management systems	●	▲	▲	▲	●	▲	–	▲	–
Social network satellites	▲	●	▲	▲	▲	●	●	▲	●
Public and password-protected areas	●	▲	●	●	▲	●	▲	▲	▲
APIs	●	▲	●	●	●	●	●	●	–
Centralized accounts and login	●	▲	●	●	●	●	–	▲	●

With an understanding of the functionality required for a community and the functionality available, someone initiating a community needs to set up a tool or set of tools. To oversimplify, there are two alternatives to consider: a single all-in-one community platform or a “best of breed” approach that brings together a range of excellent tools for each specialty into a loose package that the community uses. In reality, this is somewhat of a false dichotomy because it is almost impossible to find *all* desired services in one package and unnecessarily unwieldy not to have *some* core platform to be the “home” site. At this point it is best to plan for a home platform that serves as a core to the community and then some additional elements as it makes sense to add them.

The Core Platform

As noted earlier, online communities need to focus on doing a small number of things really well, rather than offering a variety of services. In order to create this focused, simple experience, having a core platform makes sense. This platform will contain the major functionality the community will use, as well some additional supporting features to augment the major pieces. There are three main benefits to having a core platform as the basis for the community:

1. Having a clear, first place to go provides a **seamless user experience**. Users do not need to keep track of several URLs and site layouts; they learn one system and can do much of what they need.
2. Fewer systems makes **community management easier**. Those initiating a community can develop 2 or 3 main ways for users to interact—tailored to the community goals and user needs—and then train and support members on just those features.
3. An “all-in-one” system can be a **cheap and easy pilot**. Many of the options for core platforms are delivered either as online services (“software as a service”) or as fairly easy-to-set-up downloads. They do much of what is needed “out of the box,” so a community initiator can put a platform in place with a low initial investment and begin generating use. The community managers will then typically have the option to configure the platform in at least a few and sometimes a multitude of ways.

Common features provided in most core platforms are: document/file repository, blog/announcements area, member profiles, general information area, commenting and discussions, and some form of calendar. Many will offer more, but this list is a common core. Most allow administrators some measure of control over which features appear,

allowing a community manager to turn on a limited set of features at launch and then add others over time. Below are further examples.

Example Core Collaboration Platforms

Community leaders and stakeholders can choose from among numerous core online collaboration platforms. (Wikipedia maintains one list (over 250 platforms) at http://en.wikipedia.org/wiki/List_of_collaborative_software.) Although it is possible to group these platforms into categories (“workspace products,” “conferencing products,” “community products,” etc.), each platform has unique characteristics that complicate straightforward classification. There are several important characteristics that need to be considered when selecting a primary platform for a given online community. Three of the most important defining qualities would include the following:

- **Feature Set**—Collaboration platforms vary in how they aggregate different built-in features such as discussion boards, blogs, calendars, and document libraries. Some platforms are feature-rich, while others are purposely simplistic. There is often a tradeoff between having many feature choices, often accompanied by complexity that causes confusion, and features that are limited but easier to understand and use.
- **Open-Source**—Collaboration platforms have emerged from both open-source and commercial environments. Open-source platforms often carry the advantages of no licensing costs, full access to the code base, multiple options for commercial implementation and management assistance, and, sometimes, an active international community of developers supporting the product. On the other hand, the platforms can be rough around the edges, may require programming skills to maintain, and may lack built-in support services. In contrast, the best proprietary platforms can be smoothly functioning and supported but bring inflexibility in coding changes and at times carry a high licensing cost.
- **Hosting**—Collaboration platforms can be provided in a hosted environment—often referred to as software as a service (SAAS)—or as a software product you host yourself. SAAS platforms offer ease of setup and configuration, are updated automatically, and generally have proven scalability. Software solutions by contract can offer more flexibility and control. Both SAAS platforms and software solutions can vary greatly in price, from free to six figures per year for a unique instance.

Against this backdrop of features and tradeoffs, the following list describes a representative selection of popular platform choices:

- **Jive SBS**—Jive SBS is a full-featured, proprietary platform that supports online community interactions of many types. Without any configuration, it has user-friendly functionality for posting to discussions and blogs, commenting on content, and updating user profile pages. It allows users to create document

libraries and tag documents. Administrators can also group tags to provide a level of hierarchy to larger libraries (e.g., topics and subtopics). Jive is a commercial platform and can be expensive for larger communities.

- **Drupal Commons**—Drupal Commons is a versatile and complete open-source community platform. It approximates the “out of the box” functionality of Jive, integrating a significant suite of tools, handling content and user permissions well, and allowing customization to unique needs. The tool is provided by Acquia, the enterprise technical services company that supports Drupal, one of the largest and best established open-source content-management systems. As such, a large, global, well-organized, and extremely active developer community effectively backs it.
- **Central Desktop**—Central Desktop is a proprietary, hosted platform that does a number of things well. It ably supports collaborative workspaces that involve task and document management. It has excellent integration with e-mail, allowing members to both push e-mail notifications when they publish content and allow users to cc: workspaces when they send e-mails. It also provides fairly advanced milestone and task-management functionality, useful for task and project management.
- **SharePoint**—SharePoint is a proprietary platform owned by Microsoft that can be used as software or in a hosted environment. Like Central Desktop, SharePoint is relatively strong for project and document collaboration. It is flexible in terms of taxonomies, supporting multiple facets of metadata for large document repositories. It also has strong integration with MS Office files.
- **BuddyPress**—BuddyPress is an open-source social networking software suite built upon the popular blogging platform WordPress. It provides many of the common features expected in social networking platforms, with the advantage of using a set of technologies familiar to a global Web development community. BuddyPress itself is relatively new (May 2009) and, as such, is evolving quickly. The underlying WordPress platform, however, is well established and has an active developer community.
- **Edmodo.com**—Edmodo is a different option from others on this list in that it is a custom-built education social networking system rather than a general collaboration product. Its focus is on providing a safe, efficient means for teachers to have online collaboration in their classrooms and districts to have a larger collaboration infrastructure. Within this framework, however, there is also a substantial teacher-to-teacher community area, and this feature could certainly grow to include administrators or other educational professionals. The tool provides social networking functionality as well as discussions and document sharing. The interface is straightforward and borrows heavily from Facebook, making it easy to adopt. It is currently free for all users.
- **PBWorks**—PBWorks is a commercial, hosted wiki-based platform. Wikis are collaboratively editable tools that allow a group of users to easily set up pages

and link to subpages, attach documents, and have discussions about those pages. Although it does have a “classroom” pricing plan that is reasonable, it is generally designed for enterprise-level users and is, for that reason, somewhat more expensive.

- **Groupsite**—Groupsite is a hosted workspace product that is easy to implement and is intuitive to use. It offers sufficient themes and configuration options to meet most organizations’ needs. It provides nice integration with e-mail, with easy-to-configure notifications tied to discussions, announcements, and other interactive features.
- **Ning**—Ning is a hosted workspace product that boasts the largest numbers of users and networks of all the private social networks. It is reliable, scalable, and relatively configurable, including through incorporation of external applications that support its API.
- **SocialGO**—SocialGO is a social networking platform that integrates online collaboration with social networking. It also includes an online video chat feature. It is simple to set up with a step-by-step process for editing the site. Users can send out bulletins to all other users. Although the free basic version is usable, most of the better features require a paid subscription.
- **Tomoye Community Software and Social Sites**—Tomoye Community Software is a proprietary platform owned by Newsgator. It is a social networking and collaboration platform built specifically for communities of practice. It aggregates multiple features like document sharing, blogs, wikis, videos, and Q&A. It has a mechanism that allows users to interact with the site via e-mail, thus making the site more accessible. It showcases community experts as a function of their contributions to the community, displaying whether they are helpful, active, highly networked, and so on. The platform also allows users to rate content highlighting what members consider to be the most valuable content for the community. The Social Sites version of the software is designed to integrate with SharePoint. Like Jive, it can be quite expensive for large communities.
- **Alfresco Share**—Alfresco Share is an open-source community platform, built to allow collaboration on content and projects with a globally dispersed team. It includes social features, such as status changes, tagging, and activity feeds. It has collaboration tools that include a document library, blog, wiki, calendar, workflow, and task-management feature. Through a personalized dashboard, users can view the activities across all the sites to which they belong. The sites also have a built-in reader that allows users to view multiple types of documents within the application without the need to download them. It is easy to set up, customize, and scale.

Supporting Services

Almost invariably, some features that the community needs will not be available as part of the selected core platform. A community may need some of these early in its life for particular needs (e.g., a webinar tool to support a community with many offline discussions), but many others may need to be added over time. After running a successful community based on a document repository and a discussion area, for instance, a community may decide to use a social bookmarking service to gather links from across the Web. The community may also want to survey members, in which case a separate survey tool would be useful.

As a community grows, it can explore a wider set of technologies. Are there incentive tools that can be integrated to spur growth and activity? Can the team leverage an external wiki or collaborative authoring tool to create a shared knowledge product? Should leaders of the community launch a blog to highlight the community's work? In these and many other ways, the community can and should expand beyond the core platform as its needs evolve.

These examples represent cases where a community grows such that it still appears to be a single destination, but it is actually a loosely (at best) connected set of functions. APIs, feeds, widgets, and other integration tools can become an avenue for stitching services more explicitly. Some of this work is relatively easy within the tools in question, but other integrations are harder. In fact, there is an inevitable trade-off between the value of having a more connected, coherent system and the cost of the required integration work. The hard work of a higher level of integration work may be justified, though, if re-creating the more singular experience of the “early days” translates into more efficient work, further growth, or new and productive kinds of interaction.

Third-Party Sites and Social Media

Social media sites represent a special kind of supporting service because they can play multiple kinds of roles. They provide a major asset in their sheer size and, if used well, should at least be part of a marketing strategy for a new community. Those starting a new community focused on Topic X ought to put out notifications on Facebook and Twitter, for instance, if the community is quite public, or perhaps LinkedIn, if it needs to be limited to certain audiences.

Social media sites can also serve as an ongoing dissemination channel for communities. Some communities can use feeds to social media sites as a way to broadcast content to members and target audiences who prefer to interact through that channel (or who are simply less likely to visit the core site).

These services can be central aspects of the community in their own right. Twitter recently united protesters in Tunisia and Egypt. LinkedIn can serve as a discussion-based community for professionals of many varieties. Even Facebook can be used in this capacity (though many users think of it as a “play” space and not a “work” space). No

such community service provides a comprehensive enough experience to be a true core platform in most cases, but, if audiences are already using one well in another context, incorporating these services into the community architecture could be valuable. Community designers might also consider creating custom applications that extend these systems in a manner that serves the community's goals.

A major challenge is to decide which specific activities should remain "on-domain" and which should take place on the social media sites. As a general rule, social media site communities tend to be more casual but have broader reach. For that reason, they serve as a good catchment from which to identify and migrate active users to your core community. The Obama campaign, for example, found that their huge Facebook communities did not result in significant donations, but users who migrated from Facebook to Obama's dedicated community sites (e.g., barackobama.com) did donate. Social media communities also represent an excellent communications outlet to amplify community efforts. Many organizations set staffing targets of level of effort "on-domain" and "off-domain"; there is no set rule, but it is not unusual to divide time evenly between the two.

Conclusions

Implementing online community of practice technology has its challenges, but, thankfully, clear roadmaps exist for success. Key points detailed in this summary follow:

- Take the time to map out goals and scenarios for what target audiences need to do in the community. Think about the purpose of the community. Let these considerations guide and inform technology selection.
- Ensure there are resources available for community management tasks such as training, user support, and metrics monitoring.
- Plan to start with a small set of features in a “pilot” setting. Give this narrowly scoped community great attention (i.e., do those few things well).
- Allow the community to grow in unexpected ways that are goal- or purpose-consistent. Successful communities do this. Honor this natural process.
- Expect the community to evolve over time. Add new features only when there is a demonstrable need.
- Plan to have a “home base” platform but also expect that using other tools alongside it will be valuable. Especially early on, don’t spend energy on making multiple systems flow perfectly together; if a concept is useful, fine-tune it later.

References

Cambridge, D. (2010). *Eportfolios for lifelong learning and assessment*. San Francisco: Jossey-Bass.

Wenger, E., White, N., & Smith, J. D. (2009). *Digital habitats: Stewarding technology for communities*. Portland, OR: CPsquare.

Appendix: Example User Scenarios

Below are some examples of user scenarios from an education-related community project (they have been genericized):

- Scenario 1: [Audience 1] posts meeting minutes on [name] workspace.
- Scenario 2: [Audience 1] posts response to [Audience 2] question.
- Scenario 3: [Audience 1] synchronizes team calendar with personal calendar.
- Scenario 4: [Audience 1] is traveling and needs off-line access to documents.
- Scenario 5: [Audience 1] creates internal wiki and resources on frequently asked questions (e.g., not available to grantees).
- Scenario 6: [Audience 1] posts question or comment to internal discussion or blog (e.g., not available to [Audience 2]).
- Scenario 7: [Audience 1] needs to access official/latest template.
- Scenario 8: [Audience 2] wants to suggest new user that needs to be added.
- Scenario 9: [Audience 2] wants to anonymously post to a discussion/feedback.
- Scenario 10: [Audience 2] cannot remember his or her password and wants to reset it.
- Scenario 11: [Audience 2] wants to change his or her notification preferences or modify his or her profile.
- Scenario 12: [Audience 2] wants to identify a need for new content and make a suggestion for the curated repository.