2008 Proposed Product Management Process

⚠ **New Process Suggestion**

- This is an idea for discussion, please feel free to edit, comment, etc.

Current State

The current requirements process in Sakai, while it results in valuable work getting done, has a number of issues:

1. It operates almost exclusively from feature requests submitted in Jira, which leaves out many sources of good ideas that could be added to Sakai.
2. The items that come out of the process do not necessarily have development resources dedicated to working on them. This is frustrating for those who worked on the process and those who voted on the items.
3. It doesn't result in a coherent roadmap for Sakai that can be communicated to the community (and beyond).

I'm proposing that we either abandon this process or, at the very least, add another process that I will call the "Sakai Product Management" Process.

Goals for Sakai Product Management Process

The major goals I would have for this process are as follows:

1. To produce a roadmap for the next "Major Release" of Sakai (see the proposed release practices for what this means). Generally, the time line for publishing this roadmap will be 6+ months prior to the major release.
2. For the community to have a reasonably high degree of confidence that most of the items on this roadmap will be substantially complete by the next major release. This is purposefully squishy. Roadmaps often don't come to fruition, even in commercial enterprises, so I would expect a certain amount of slippage or feature reduction along the way.
3. To assist in combining like efforts towards a common goal. We need a formal mechanism for identifying common interests in advance of when resources will become available.
4. To increase visibility about what is being worked on.

Some beneficial effects I would also hope to come out of this would include:

- The ability to conceive of and plan larger, longer-term projects.
- Increased cross-institutional tool/code development.
- A broader set of use-cases informing development, hopefully leading to a better user experience.
- Longer development timelines, allowing for fuller functional requirements development, more formal usability design and more flexible/abstracted code.

Proposed Process

The general approach I have in mind is:

1. A Sakai Foundation Product/Project Manager would gather ideas from jira, community interviews and competitive analysis. He would create a list of approximately 15 "medium - large" sized projects targeted for the next major release.
2. This list would be circulated to the community for comment and would be roughly scoped by technical & functional experts as well as designers in the community. The goal is to give the group a rough estimate of how big each item is relative to the others.
3. This list is brought to a "product management committee" (or some such) made up of a small number of folks (primarily, but not exclusively) from schools who will be contributing development resources over the next period of time. This group sets the top few priorities and commits to working towards them, ideally with some collaboration between institutions. The foundation kicks in resources where necessary/possible. (Examples of the level I'm thinking about: Common Cartridge support, site/user hierarchy, user interface overhaul of "basic" tools, Improved goal awareness, Improved group awareness...generally things that will take 4+ months to complete)
4. These priorities (probably 3-5 items from the list) become our published roadmap for Sakai, of course filled in with all the little things that everyone is working on anyways. They would then get more formally designed and scoped and a schedule would get built. Ideally, each of the items becomes a significant part of the "minor" releases envisioned in the release practices proposal. These aren't commitments, but goals. Others around the community are invited to join in the detailed requirements development process around these priority areas.
5. The committee meets every 6-8 weeks to review progress and discuss what might have changed since the last meeting.

Committee members

The reason for doing this in a small committee instead of simply "on the lists" is (a) to allow a meaningful interactive discussion between folks who have resources and (b) to not have people voting on things that aren't going to get done. This latter things is very frustrating for everyone. I'd rather acknowledge the fact that the work is going to be based on the priorities of the groups with development resources (from design through QA) and put a process in place that collects feedback from everyone but doesn't pretend that those things are going to be what happens, necessarily.

The reason for having a small committee instead of just letting things continue as they are is that I believe (a) we need to start looking further out than it appears to me we have been and (b) we can do a better job of identifying common priorities and lining up resources behind those.

Selection of the committee is problematic and has the risk of being a "star chamber". But, at the end of the day, we should acknowledge the fact that those with resources to commit are going to have the largest say in what happens. Explicitly naming those folks should increase visibility into the process and create a place to provide input. There are a couple of approaches to selecting this group. The one I'm currently favoring is to get an initial group from those schools who are committing "significant" resources for the next major release and have them invite a few others as they see fit.

The mix of skills on the committee is also important. The most important, I think, is some expertise in teaching and learning with technology (functional expertise). Still, there needs to be enough technical skill in some members of the group to understand the general technical approach being proposed for a certain roadmap item. At the risk of naming names, people like John Norman, Clay Fenlason and Stephen Marquard (to pick one person from each of 3 continents) have a good mix of both functional and technical expertise. But we should **not** be afraid of a few people without technical expertise and a few without teaching/learning expertise - the overall mix is what is important. I also think it is good to include one or two folks from schools without resources - we need that voice if we want to produce a product that is widely used.

There is a risk of this group getting too big for a meaningful meeting. That's a high-class headache, in my view, and if it comes to that we can somehow split the group. We may already have two groups...OSP is already doing something like this. I would imagine they would keep their process focused on portfolios.

In any case, that's enough for now...please add your comments.