**Are We Open Yet?**

Why might your team want to move into open working? And how might you do so? Great piece from Matt reporting on a Mozilla meet up, letting you and your team prepare to shift yourselves into more open working.

**Why might you team want to move into open working?**

For greater results, speed, agility, energy, creativity, quality, resource use, participation and to punch above your weight.

The goal of open is NOT:

* public performance: creating the fake appearance of consultation.
* endless opinion-sharing: never-ending “feedback”, diversion into trivia
* magic “crowd-sourcing”: crowds aren’t smart — communities of peers are.

**What can block you?**

It does not happen because someone stamps their feet and says be open! Colleagues typically have all sorts of excuses, reasons, fears and objections such as:

* **“I’m too busy”.** I want to blog and surface my work more publicly, but somehow it always falls to the bottom of my to-do list.”
* **“People can be mean”.** Opening our work to criticism from others is often scary.
* **“We’re not ready”.** It doesn’t make sense yet to open our project up to participation and public attention — before we have the tools in place to meaningfully absorb that participation and attention. This can be valid. But while you may not be ready for participation at scale — you probably *are* ready for some early testing, prototyping, and smart co-building from colleagues.

Which takes us to…

**Tip: Agree on what gear of open you’re working in.**

Working open is more of a slider or dial than an “on/off” switch. At a given point in time on a given project, you might collectively agree to work in a range of different gears or levels of open. At Mozilla it feels like we’re often in one of four gears:

**0) Closed:** For example, reasons can include anything involving personal data, security, etc.

**1) “Not yet, but soon.”**  We want to work open — but we’re not ready yet. We’re not ready for widespread attention. Or can’t meaningfully absorb offers from people to help. So let’s wait until we are.This is a totally reasonable gear to operate in — but can also become a trap or semi-permanent holding pattern. Without forcing functions or test cases, it’s a recipe for going slow.

**2) “Open”.** Our standard default setting. Working in public spaces like etherpads and community lists, instead of closed email threads.  Sharing signposts, drafts, prototypes and roadmaps on our blogs, etc. The primary goal is surfacing what’s needed to enable smart co-building. If we don’t, not only will our communities have no idea how to get involved — our immediate peers and colleagues won’t be able to help as effectively, either.

* This gear is like speaking out loud in a normal voice — not shouting or using a megaphone. It isn’t the same as shouting from the rooftops, issuing a press release, or getting your picture on the cover of *Rolling Stone*.

**3) “Shout it from the rooftops”.** Taking it up a notch to a higher order of magnitude. Like: “Holy crap we’re releasing Firefox 4!” Or: “We’re ready for the cover of *Rolling Stone*!” Participation at scale. From co-builders to more mainstream participants or consumers.

**Why is having clear consensus on what mode you’re in useful?**

Many of us are uncomfortable about the distinction between gears 2 and 3. It’s easy to confuse “working open” with: “let’s tell the whole world about it.”  We worry that saying something in a blog post or etherpad will commit us to things in public before they’re ready. When in fact, those fears rarely — if ever — materialize. And the benefits far outweigh the imagined risk.  Working closed, slow or overly cautious is a silent killer that saps momentum.

**Tip: Testing and rapid prototyping are different from “asking for feedback.”** Putting a new design in front of someone and quietly observing how they interact with it is testing. Putting a design in front of someone and asking: “So… what do you think?” is asking for feedback. When you ask for feedback in open channels, you never know what you may get back — people are busy, and can’t always offer thoughtful opinions just because you ask for them.

Testing a new design, draft or prototype, for example, tells you where it’s working, where it’s surprising, and where it’s broken. And while you may not agree with all the feedback you get, you will always learning something from releasing early and often in public. Feedback is sometimes useful. Testing always is.

**Tip: Random crowds of people offering casual opinions isn’t always helpful**. There’s a popular misconception that the value of working in the open is that transparency enables the wisdom of crowds to constantly offer feedback and new points of view. We don’t think that’s really it. Testing early and often always is always helpful.

**Tip: The main value of working open is reducing transaction cost, administrivia, and collaborative friction with smart communities of peers.** That’s different than over-relying on the casually offered opinions of whoever shows up, or waiting for the crowd to do your work or solve your problems for you. 

**Please share your ideas, tactics or success stories for working in the open here**. As Mozilla Drumbeat grows to work with newcomers in new spaces, we’re constantly encountering people who are curious about what working open means and how Mozilla can help them do it. We see it as something we’re constantly teaching each other how to do. So please share.

[](http://openmatt.files.wordpress.com/2011/04/jump-open-ftw.jpg)

Brave Mozillians leap into the open void.

***Why do we work in the open?***

***How do you do it?***

***In what ways is it scary or difficult?***

The key question is not whether every itty bitty piece of communication or decision-making should be “open” or “closed.” The key question is:

**How does working in the open enable useful participation?**

**How does it help us be more agile?**

**How does it produce visible progress and momentum?**

**How does it help us do good?**

**The goal of open is:**

* **participation**. rocket fuel for smart collaboration.
* **agility**. speed. flexibility. getting shit done.
* **momentum**. communities want to push boulders that are already rolling.
* **testing and rapid prototyping**. iterating and refining as we go.
* **leverage**. getting greater bang from limited resources. punching above our weight.

[](http://www.flickr.com/photos/nooooddy/120351180/)

That boulder isn't going to roll itself.

**Acknowledge valid obstacles and fears.**

**Stamping our feet and saying, “work open, damnit!” doesn’t work.** The goal is pragmatism, not fundamentalism, especially when we’re working with new communities and partners with diverse work cultures and backgrounds. Some of the common fears or concerns we hear at Mozilla Drumbeat:

* **“I’m too busy.”** “I want to blog and surface my work more publicly, but somehow it always falls to the bottom of my to-do list.”
* **“People can be mean.”** Opening your work to criticism from others is often scary.
* **“We’re not ready.”** This can be valid. It doesn’t make sense to open your project up to participation and public attention — before you have the tools in place to meaningfully absorb that participation and attention. But while you may not be ready for participation at scale — you probably *are* ready for some early testing, prototyping, and smart co-building from colleagues. Which takes us to…

**[](http://openmatt.files.wordpress.com/2011/04/what-gear-of-open-are-you-in.jpg)**

**Agree on what gear of open you’re working in.**

**Working open is more of a slider or dial than an “on/off” switch.** At a given point in time on a given project, you might collectively agree to work in a range of different gears or levels of open. For Mozilla Drumbeat projects, it feels like we’re often in one of four gears:

**0) CLOSED**

For example, anything involving personal data, security, etc.

**1) “Not yet.”**

SOON. We want to work open — but we’re not ready yet. We’re not ready for widespread attention. Or can’t meaningfully absorb offers from people to help. So let’s wait until we are.

This is a totally reasonable gear to operate in — but can also become a trap or semi-permanent holding pattern. Without forcing functions or test cases, it’s a recipe for going slow.

**2) “Open”**

**Our standard default setting.** Working in public spaces like etherpads and community lists, instead of closed email threads.  Sharing signposts, drafts, prototypes and roadmaps on our blogs, etc. The primary goal is surfacing what’s needed to enable smart co-building. If we don’t, not only will our communities have no idea how to get involved — our immediate peers and colleagues won’t be able to help as effectively, either.

At the same time, working open isn’t the same as shouting from the rooftops, issuing a press release, or getting your picture on the cover of *Rolling Stone*. This gear is like speaking out loud in a normal voice — not shouting or using a megaphone.

**3) “Shout it from the rooftops”**

Like: “Holy crap we’re releasing Firefox 4!” Or: “We’re ready for the cover of *Rolling Stone*!” Taking it up a notch to a higher order of magnitude. Participation at scale. From co-builders to more mainstream participants or consumers.

**Why is having clear consensus on what mode you’re in useful?** Many of us are uncomfortable about the distinction between gears 2 and 3. It’s easy to confuse “working open” with: “let’s tell the whole world about it.”  We worry that saying something in a blog post or etherpad will commit us to things in public before they’re ready. When in fact, those fears rarely — if ever — materialize. And the benefits far outweigh the imagined risk.  Working closed, slow or overly cautious is a silent killer that saps momentum.

[](http://www.flickr.com/photos/geraldbrazell/4562022876/sizes/m/)

Are we open yet?

**Testing and prototyping vs. “asking for feedback”**

**Testing and rapid prototyping are different from “asking for feedback.”** Putting a new design in front of someone and quietly observing how they interact with it is testing. Putting a design in front of someone and asking: “So… what do you think?” is asking for feedback. When you ask for feedback in open channels, you never know what you may get back — people are busy, and can’t always offer thoughtful opinions just because you ask for them.

**Random crowds of people offering casual opinions isn’t always helpful**. But testing early and often always is. I think there’s a popular misconception that the value of working in the open is that transparency enables the wisdom of crowds to constantly offer feedback and new points of view. I don’t think that’s really it.

**The main value of working open is reducing transaction cost, administrivia, and collaborative friction with smart communities of peers.** That’s different than over-relying on the casually offered opinions of whoever shows up, or waiting for the crowd to do your work or solve your problems for you. 

**Feedback is sometimes useful. Testing always is.** Testing a new design, draft or prototype, for example, tells you where it’s working, where it’s surprising, and where it’s broken. And while you may not agree with all the feedback you get, you will always learning something from releasing early and often in public.